

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

PRÉVISION DU TRAFIC INTERNET - MODÈLES ET APPLICATIONS

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN INFORMATIQUE

PAR

MOHAMED FATEN ZHANI

JUIN 2011

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Je tiens à adresser mes sincères remerciements et à exprimer toute ma gratitude à mon encadrante Pr Halima Elbiaze. Elle m'a bien dirigé et aidé tout au long de l'élaboration de cette thèse et elle m'a soutenu scientifiquement, financièrement et personnellement. Son enthousiasme, son dynamisme et surtout sa patience m'ont à chaque fois permis de rebondir dans les moments difficiles. Je tiens à lui exprimer ma reconnaissance.

Je remercie aussi Pr Farouk Kamoun qui m'a également soutenu et qui a suivi de près toutes mes recherches. Le professeur Farouk Kamoun m'a prodigué son expérience me guidant jusqu'à ce jour. Il a fait preuve de beaucoup de rigueur et de gentillesse et j'espère qu'il continuera à me guider au-delà de cette thèse. J'admire extrêmement ses qualités humaines d'écoute, de compréhension et de direction.

Je ne sais pas si ces quelques mots seront à la hauteur des remerciements que ces deux professeurs méritent, mais j'espère que cette thèse sera un remerciement suffisant au soutien et à la confiance sans cesse renouvelés dont ils ont fait preuve à mon égard.

Je remercie particulièrement mon ami Pr Wael Hosny Fouad Aly, professeur à l'académie des sciences et des technologies en Egypte. Il m'a sans aucun doute transmis, au travers des centaines d'heures et de nuits de travail passées ensemble, une grande partie de sa motivation, son dynamisme, sa détermination et ses idées brillantes.

Je remercie tous ceux sans qui cette thèse ne serait pas ce qu'elle est, aussi bien par les discussions que j'ai eu la chance d'avoir avec eux que par leurs suggestions ou encore leurs contributions. Je citerais Pr Wessam Ajib, professeur à l'Université du Québec À Montréal (UQAM), Pr Omar Cherkaoui, professeur à l'UQAM, Pr Faten Rouai, professeur à la faculté de Nabeul, Pr Chadi barakat, chercheur à l'INRIA au centre de recherche Sophia Antipolis en France, Pr Christophe Diot, directeur du centre de recherche Thomson à Paris, Pr Philippe Owezarski, chercheur au laboratoire LAAS à Toulouse en France et Pr Augustin Soule, chercheur au centre de recherche Thomson à Paris.

Je n'oublie pas de remercier tous mes collègues au laboratoire LTIR à l'UQAM et au laboratoire CRISTAL à l'École Nationale des Sciences de l'Informatique à Manouba (ENSI). Je citerais Iyas Khayata, Mouna Ayari, Hela Boucetta, Salem Belhaj avec qui j'ai travaillé et partagé des durs moments.

Je remercie le président du jury Pr Mounir Boukadoum, professeur à l'UQAM et les membres du jury Pr Guy Bégin, professeur à l'UQAM, et Pr Roch H. Glitho,

professeur à l'université de Concordia. Ils m'ont fait l'honneur d'être les rapporteurs de cette thèse. J'éprouve un profond respect pour leurs travaux et leurs parcours, ainsi que pour leurs qualités humaines. Le regard critique, juste et avisé qu'ils ont porté sur mes travaux ne peut que m'encourager à être encore plus perspicace et engagé dans mes recherches.

Je remercie chaleureusement mon amie de cœur Fatine Kahli pour son soutien et son amour. Je remercie également mes amis en Tunisie et au Canada. Je citerais du Canada : Haythem Kefi, Mohamed Amira, Tarek Moufaker, Nihed Ben Salah, Usef Faghihi, Mounir Arab, Mahdi Driouch et Abdelhalim Driouech. Je citerais de la Tunisie : Mohamed Abbassi, Khaled Abda, Walid Atrous et Omar Belkhodja.

Je tiens à remercier toute ma famille Zhani et Qutub. Je remercie évidemment mon cher père Abdelaziz Zhani et ma chère et tendre mère Nerimen Qutub et mes deux frères Fares et Firas pour leur irremplaçable et inconditionnel soutien. Ils sont toujours présents pour écarter les doutes, soigner les blessures et partager les joies. Je sais que ces quelques mots ne sont pas suffisants pour les remercier, mais je leur dédie cette thèse et j'espère qu'ils trouveront dans ce travail un simple témoignage de ma gratitude et de ma reconnaissance.

Je n'oublie pas de remercier tous mes enseignants à l'Université du Québec à Montréal et à l'école nationale des sciences de l'informatique à Tunis pour le savoir-faire qu'ils m'ont généreusement inculqué.

Je remercie aussi toutes les personnes qui ont participé de près ou de loin au bon déroulement et la réalisation de cette thèse.

ZMF

TABLE DES MATIÈRES

LISTE DES FIGURES	ix
LISTE DES TABLEAUX	xiii
LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES	xv
RÉSUMÉ	xix
INTRODUCTION	1
0.1 Problématique	3
0.2 Contribution de la thèse	5
0.2.1 Analyse de l'effet des paramètres du modèle sur la performance de la prévision	7
0.2.2 Proposition d'une technique d'échantillonnage adaptée à la prévision du trafic	8
0.2.3 Proposition d'un nouveau mécanisme de gestion de files d'attente basée sur la prévision	8
0.2.4 Application de la prévision du trafic dans le cas des réseaux OBS . .	9
0.3 Organisation de la thèse	10
CHAPITRE I	
ÉTAT DE L'ART	13
1.1 Modélisation du trafic Internet	14
1.2 Les techniques de prétraitement des données	15
1.2.1 Les méthodes de lissage	16
1.2.2 Traitement logarithmique	17
1.2.3 Les techniques d'échantillonnage	18
1.3 Analyse du trafic Internet	20
1.3.1 Analyse spectrale (recherche des périodicités)	20
1.3.2 Analyse des dépendances temporelles	21
1.4 Caractéristiques du trafic Internet	22
1.4.1 Non-stationnarité	23

1.4.2	Auto-similarité et dépendance à long terme	24
1.4.3	Distribution à décroissance lente	27
1.4.4	La sporadicité du trafic	28
1.5	Les modèles de prévision du trafic	29
1.5.1	Le modèle autorégressif à moyenne mobile (ARMA)	30
1.5.2	Le modèle linéaire des moindres carrés (LMMSE)	31
1.5.3	Les modèles auto-similaires	32
1.5.4	Les modèles Markoviens	35
1.5.5	Le filtre de Kalman	36
1.5.6	Le modèle neuroflou α -SNF	39
1.5.7	Conclusion sur les modèles de prévision	42
1.6	Revue de la littérature	43
1.6.1	Prévision du trafic	44
1.6.2	Les mécanismes de gestion des files d'attente	49
1.7	Les réseaux optiques à commutation de rafales	54
1.7.1	Architecture du réseau OBS	54
1.7.2	Assemblage des rafales	56
1.7.3	Signalisation et réservation des ressources	56
1.7.4	Contention et perte de paquets	59
1.8	Conclusions	61
CHAPITRE II		
EFFET DES PARAMÈTRES DU MODÈLE SUR LA PERFORMANCE DE LA PRÉVISION		65
2.1	Méthodologie de la prévision du trafic	66
2.1.1	Méthodologie	66
2.1.2	Critères de performance	68
2.2	Analyse des traces d'un réseau réel	69
2.2.1	Présentation des traces et prétraitements	70
2.2.2	Analyse des traces	71
2.3	Analyse des différents paramètres du modèle de prévision	74

2.3.1	Choix des variables d'entrée du système	74
2.3.2	Choix de la granularité du trafic	80
2.3.3	Choix de la taille des données d'apprentissage	81
2.3.4	Prévision du trafic en utilisant des variables exogènes	84
2.4	Conclusion	87
CHAPITRE III		
PROPOSITION D'UNE NOUVELLE TECHNIQUE D'ÉCHANTILLONNAGE		89
3.1	Objectifs de l'échantillonnage	90
3.2	Présentation de la technique d'échantillonnage basé sur le maximum (MBS)	91
3.2.1	Preuve mathématique de la conservation de l'auto-similarité	92
3.3	Résultats expérimentaux	97
3.3.1	Présentation des traces et prétraitement	97
3.3.2	Comparaison de la technique MBS avec les techniques existantes	97
3.3.3	Exploitation des techniques d'échantillonnage pour la prévision du trafic	101
3.4	Conclusion	103
CHAPITRE IV		
PROPOSITION D'UN MÉCANISME DE GESTION DE FILES D'ATTENTE BASÉE SUR LA PRÉVISION		105
4.1	α -SNFAQM	106
4.1.1	Principe de fonctionnement	107
4.1.2	Algorithme	108
4.2	Étude analytique	111
4.3	Évaluation de performance	125
4.3.1	Simulation du premier scénario	126
4.3.2	Simulation du deuxième scénario	132
4.4	Conclusion	136
CHAPITRE V		
APPLICATION DE LA PRÉVISION DU TRAFIC DANS LE CAS DES RÉSEAUX OBS		139
5.1	Modèle proposé pour le contrôle du taux de perte	140

5.1.1	Description du modèle de contrôle proposé	140
5.1.2	Modélisation du réseau OBS et conception du contrôleur	143
5.1.3	Amélioration du modèle avec la prévision TCP	148
5.2	Évaluation de performance	154
5.2.1	Contrôle du taux de perte avec feedback et sans QoS	154
5.2.2	Contrôle du taux de perte avec feedback et avec QoS	159
5.2.3	Contrôle du taux de perte sans feedback avec prévision du trafic TCP et sans QoS	163
5.3	Conclusion	166
	CONCLUSION	169
	APPENDICE A	
	RAPPEL SUR LES PROCESSUS ALÉATOIRES	173
	APPENDICE B	
	LE MÉCANISME <i>RANDOM EARLY DETECTION</i> (RED)	177
	APPENDICE C	
	FONCTION DE TRANSFERT D'UN SYSTÈME ASSERVI	179
	APPENDICE D	
	STABILITÉ ET REPRÉSENTATION DES LIEUX DES PÔLES	181
	BIBLIOGRAPHIE	185

LISTE DES FIGURES

Figure	Page
1.1 Illustration des trois techniques d'échantillonnage (He et Hou, 2005). . .	20
1.2 Exemple d'un réseau de neurones (trois règles et deux entrées).	41
1.3 Mécanisme RED.	49
1.4 Architecture d'un réseau OBS.	55
1.5 Réservation immédiate (JIT).	57
1.6 Réservation différée.	59
2.1 Méthodologie de la prévision.	68
2.2 Durée des connexions TCP (Auckland-VIII).	71
2.3 RTT des connexions TCP (Auckland-VIII).	71
2.4 Distribution des tailles des paquets pour le trafic Auckland.	73
2.5 Distribution des tailles des paquets pour le trafic CESCA.	74
2.6 Information mutuelle par rapport au nombre de retards et à la granularité (données Auckland-VIII).	77
2.7 Coefficient de corrélation par rapport au nombre de retards et à la gra- nularité (données Auckland-VIII).	77
2.8 (a) Information mutuelle et (b) coefficient de corrélation pour différentes granularités (données Auckland-VIII).	78

2.9	Erreur de la prévision par rapport au nombre de retards et à la granularité (modèle ARMA, données Auckland-VIII).	79
2.10	Erreur de la prévision (modèle ARMA, données Auckland-VIII).	80
2.11	L'écart-type et l'erreur de prévision (RMSE) par rapport à la granularité.	80
2.12	Erreur de la prévision par rapport à la taille des données d'apprentissage (données Auckland-VIII, granularité 1 s).	82
2.13	Pourcentage des différents ensembles par rapport au trafic total.	85
2.14	Résultats expérimentaux en utilisant les variables exogènes (données Auckland-VIII).	86
3.1	Illustration de la technique MBS.	92
3.2	Moyenne des données échantillonnées.	98
3.3	Variance des données échantillonnées.	99
3.4	Comparaison entre le trafic lissé et le trafic échantillonné avec MBS.	99
3.5	Mesure de la sporadicité (PM).	100
3.6	Paramètre de <i>Hurst</i> pour différentes granularités.	100
3.7	Prévision avec le modèle LMMSE en utilisant les données échantillonnées.	102
4.1	Principe de fonctionnement du mécanisme α -SNFAQM.	107
4.2	Congestion grave et congestion légère.	108
4.3	Algorithme α -SNFAQM.	109
4.4	État futur de la file par rapport au taux de perte $q(\tau)$ et le débit d'entrée $IR(\tau)$	115
4.5	La fonction $\sigma(IR(\tau))$ ($n = 20$).	119

4.6	Probabilité de réalisation de chaque état futur en fonction du nombre de connexions.	122
4.7	Taille de la file d'attente en fonction du nombre de connexions ($th_{min} = 5$).125	
4.8	Topologie du premier scénario.	126
4.9	Charge de la file d'attente en fonction du nombre de sources.	128
4.10	Les états futurs de la file d'attente prédits par le mécanisme α -SNFAQM (10 sources).	128
4.11	Taille de la file d'attente (10 sources).	129
4.12	Taille moyenne de la file d'attente.	130
4.13	Débit moyen de perte des paquets.	131
4.14	Débit moyen de sortie de la file d'attente.	131
4.15	Débit à la réception (niveau application).	132
4.16	Topologie du deuxième scénario.	132
4.17	Débit moyen de la perte des paquets.	133
4.18	Taille moyenne de la file d'attente.	134
4.19	Délai moyen de bout en bout (entre la source 0 et la destination 0). . .	134
4.20	Débit moyen de sortie (file 2) et débit à la réception.	135
5.1	Modèle de système asservi en boucle fermée sans QoS.	141
5.2	Modèle de système asservi en boucle fermée avec qualité de service. . . .	143
5.3	Modèle de système asservi en boucle fermée - les transformées en Z . . .	147
5.4	Modèle proposé sans feedback.	149

5.5 Exemple de retransmission TCP dans un réseau ordinaire et dans un réseau OBS.	152
5.6 Topologie étudiée.	154
5.7 Débit de burstification et taux de perte des rafales entre la source s_0 et la destination d_0	155
5.8 Taux de perte des rafales.	156
5.9 Débit de burstification.	157
5.10 Taux de perte des rafales pour tout le réseau.	158
5.11 Utilisation du lien.	158
5.12 Le réseau NSFNet.	159
5.13 Taux de perte des rafales de la source s_0 à la destination d_6 (sans contrôleur). 161	
5.14 Taux de perte des rafales de la source s_0 à la destination d_6 (avec contrôleur). 161	
5.15 Utilisation du réseau.	162
5.16 Taux de perte des segments réel (mesuré) et estimé entre la source s_0 et la destination d_6 (sans contrôleur).	165
5.17 Taux de perte des rafales réel (mesuré) et estimé entre la source s_0 et la destination d_6 (sans contrôleur).	165
5.18 Taux de perte des rafales (BLR) contrôlé avec EBLR entre la source s_0 et la destination d_0	166
C.1 Système asservi en boucle fermée.	179
D.1 Exemple d'un système asservi en boucle fermée.	182
D.2 Exemple de représentation des lieux des pôles.	184

LISTE DES TABLEAUX

Tableau	Page
1.1 Résumé des différents AQM	53
2.1 Caractéristiques des traces de trafic Auckland-VIII et CESCA-I	72
3.1 Résumé des résultats obtenus	103
4.1 Notation pour chaque état	112
4.2 Caractérisation des états futurs de la file d'attente	118

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

α _SNF	Système NeuroFlou nommé α _SNF
α _SNFAQM	Active Queue Management based on the α _SNF model
ACF	Autocorrelation function
AQM	Active Queue Management
ARIMA	AutoRegressive Integrated Moving Average
ARMA	AutoRegressive Moving Average
ARV	Average Relative Variance
ATM	Asynchronous Transfer Mode
bps	bit par seconde
BLR	Burst Loss Ratio
BL	Burst Loss Rate
BR	Burstification Rate
BSS	Biased Systematic Sampling
FARIMA	Fractional AutoRegressive Integrated Moving Average
FB	Formula-Based
FBM	Fractional Brownian Motion
FDN	Fractional Differencing Noise
FGN	Fractional Gaussian Noise
FTP	File Transfer Protocol
GB	Gigabyte
Gb	Gigabit
Gbps	Gigabit per Second
HB	History-Based
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol

IP	Internet Protocol
JET	Mécanisme de réservation Just-Enough-Time
JIT	Mécanisme de réservation Just-In-Time
KB	Kilobyte
Kb	Kilobit
Km	Kilomètre
LAN	Local Area Network
LMMSE	Linear Minimum Mean Square Error
LRD	Long Range Dependence
MB	Megabyte
Mb	Megabit
Mbps	Megabit per second
MBS	Maximum-based sampling technique
MLE	Maximum-Likelihood Estimation
MMPP	Markov Modulated Poisson Process
MSE	Mean Squared Error
MTU	Maximum Transmission Unit
NLMS	Normalized Least Mean Square
NS-2	Network Simulator 2
NTB	Non-Training-Based
OBS	Optical Burst Switched
OSI	Open System Interconnection
PACF	Partial Autocorrelation function
PAQM	Predictive Active Queue Management
PM	Peak To Mean
POP	Point Of Presence
pps	Paquet par seconde
QoS	Quality of Service
RED	Random Early Detection

RMSE	Root Mean Squared Error
RTT	Round Trip Time
RTO	Retransmission TimeOut
SL	Segment Loss Rate
SLR	Segment Loss Ratio
SNMP	Simple Network Management Protocol
SpB	Segment Per Burst
TB	Training-Based
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VBR	Variable Bit Rate
WAN	Wide Area Network
WDM	Wavelength Division Multiplexing

RÉSUMÉ

Avec l'essor de la métrologie de l'Internet, la prévision du trafic s'est imposée comme une de ses branches les plus importantes. C'est un outil puissant qui permet d'aider à la conception, la mise en place et la gestion des réseaux ainsi qu'à l'ingénierie du trafic et le contrôle des paramètres de qualité de service.

L'objectif de cette thèse est d'étudier les techniques de prévision et d'évaluer la performance des modèles de prévision et de les appliquer pour la gestion des files d'attente et le contrôle du taux de perte dans les réseaux à commutation de rafales. Ainsi, on analyse les différents paramètres qui permettent d'améliorer la performance de la prévision en termes d'erreur. Les paramètres étudiés sont : la quantité de données nécessaires pour définir les paramètres du modèle, leur granularité, le nombre d'entrées du modèle ainsi que les caractéristiques du trafic telles que sa variance et la distribution de la taille des paquets.

Nous proposons aussi une technique d'échantillonnage baptisée échantillonnage basé sur le maximum (*Max-Based Sampling* - MBS). Nous prouvons son efficacité pour améliorer la performance de la prévision et préserver l'auto-similarité et la dépendance à long terme du trafic.

Le travail porte aussi sur l'exploitation de la prévision du trafic pour la gestion du trafic et le contrôle du taux de perte dans les réseaux à commutation de rafales.

Ainsi, nous proposons un nouveau mécanisme de gestion de files d'attente, baptisé α -SNFAQM, qui est basé sur la prévision du trafic. Ce mécanisme permet de stabiliser la taille de la file d'attente et par suite, contrôler les délais d'attente des paquets.

Nous proposons aussi une nouvelle technique qui permet de garantir la qualité de service dans les réseaux à commutation de rafales en termes de taux de perte. Elle combine entre la modélisation, la prévision du trafic et les systèmes asservis avec feedback. Elle permet de contrôler efficacement le taux de perte des rafales pour chaque classe de service. Le modèle est ensuite amélioré afin d'éviter les feedbacks du réseau en utilisant la prévision du taux de perte au niveau TCP.

Les mots-clés : Modélisation et prévision du trafic, techniques d'échantillonnage, gestion des files d'attente, réseaux à commutation de rafales, contrôle du taux de perte, qualité de service, l'automatique.

INTRODUCTION

L'Internet est devenu le média universel pour tous les types d'informations tels que les données, la voix, la vidéo et les applications temps-réel. Sa croissance et son évolution rapide en taille et en complexité ne permettent pas une maîtrise complète du réseau et de son comportement. Par conséquent, les opérateurs réseaux utilisent des techniques de métrologie pour la mise en place et la maintenance de leurs réseaux de communication afin d'offrir les meilleurs services à leurs clients et d'optimiser au maximum l'utilisation des ressources.

La métrologie de l'Internet, ou science de mesure de l'Internet, vise à superviser, mesurer et modéliser le trafic Internet afin d'évaluer la performance des réseaux en termes des différents critères de qualité de service (*Quality Of Service* - QoS). Elle permet, entre autres, de comprendre le fonctionnement des divers protocoles qui gèrent le réseau et d'analyser la manière avec laquelle ces protocoles interagissent entre eux.

Avec l'essor de la métrologie de l'Internet, la modélisation et la prévision du trafic se sont imposées comme des branches très importantes autant au niveau de la recherche en réseau qu'au niveau de la conception, de la mise en place, de la gestion des réseaux et de l'amélioration des protocoles (Brakmo et Peterson, 1995; Bush, 1999; Casetti, Kurose et Towsley, 1996; Janowski et Owezarski, 2010; John, Tafvelin et Olovsson, 2010; Chabchoub et al., 2010).

La prévision du trafic a été intensivement étudiée depuis la découverte de la nature *auto-similaire* et de la *dépendance à long terme* (LRD) du trafic des différents types de réseaux. En effet, il est établi que le trafic présente de forte auto-corrélation (Paxon et Floyd, 1995; Park, Kim et Crovella, 1997; Leland et al., 1994; Abry et Veitch, 1998). Bien que ces caractéristiques particulières causent la dégradation de la performance du réseau en termes de perte et de délai, plusieurs études ont prouvé

que l'auto-similarité peut être exploitée pour caractériser et prévoir le trafic afin de contrôler le réseau (Barakat et al., 2003; Owezarski et Larrieu, 2004; Papagiannaki et al., 2003b; Scherrer et al., 2006; Gao, He et Hou, 2002; He et Hou, 2003; He et al., 2002; Bermolen et Rossi, 2009; Casolari et Colajanni, 2009).

Ainsi, la prévision du trafic du réseau aide à résoudre plusieurs problématiques dans les réseaux comme, d'une part, la planification et le dimensionnement des réseaux qui permettent de mettre en place des capacités suffisantes pour assurer en permanence un service adéquat à tous les utilisateurs. D'autre part, la prévision peut être exploitée pour l'amélioration des mécanismes de contrôle de congestion et de ressources puisqu'on peut adapter les paramètres du réseau aux futures caractéristiques du trafic.

La prévision du trafic peut être classée selon la taille de l'intervalle de temps pour lequel elle est effectuée. Cet intervalle est appelé *horizon* de la prévision. Ainsi, on peut distinguer la prévision à court terme, à moyen et à long terme.

Pour la prévision à court terme, l'intervalle de prévision est de l'ordre de quelques millisecondes et peut aller jusqu'à l'ordre de quelques minutes. La prévision à court terme est, en général, réalisée en temps réel (On-line). Si on connaît les changements du trafic dans un futur très proche, la réservation des ressources sera beaucoup plus simple et précise. Ainsi, dans le cas des réseaux garantissant la qualité de service, l'allocation dynamique de la bande passante et la négociation dynamique de contrat seront accordées selon les résultats de la prévision. Si la valeur prédite de la bande passante excède celle qui est disponible, plusieurs solutions peuvent être envisagées : une bande passante additionnelle peut être réservée ou des mécanismes de contrôle peuvent être évoqués pour re-router les flux existants, réduire leur débit par négociation, ou bloquer les nouvelles demandes.

La prévision à court terme permet aussi d'améliorer *les mécanismes de gestion des files d'attente* (*Active Queue Management - AQM*). En effet, ces mécanismes utilisent le rejet aléatoire des paquets comme un moyen pour réduire la congestion au niveau des files d'attente et permet de prévenir les sources du trafic de la saturation du réseau. La

prévision du trafic peut être un moyen plus précis pour prendre les décisions de rejet des paquets afin de réduire les pertes.

La prévision permet aussi d'améliorer le mécanisme de contrôle de congestion du protocole de transport TCP (*Transport Control Protocol*). En effet, une prévision précise permettrait de prévoir la congestion et par suite, la connexion TCP peut adapter son débit en fonction de l'état futur du réseau.

D'autre part, la prévision peut être effectuée à moyen terme c'est-à-dire de l'ordre des heures, des journées et éventuellement des semaines. Dans ce cas, la prévision peut être utilisée pour aider à la décision de routage. En effet, la prévision de la charge d'un lien d'un réseau fédérateur¹ peut être utilisée pour prendre des décisions dynamique pour le routage des paquets afin d'éviter les éventuelles congestions au niveau des liens.

On peut aussi prévoir le trafic à long terme (mois ou années) afin de déterminer la tendance future du trafic qui peut être utilisée pour planifier les investissements pour les réseaux des fournisseurs de service Internet. En effet, la prévision peut aider à estimer à quel moment et dans quel lien le niveau du trafic sera important et dépassera la capacité existante. Ainsi, une prévision détaillée pour le modèle annuel du trafic permet de réaliser de meilleures décisions et des plans plus précis pour la mise à niveau des équipements du réseau.

0.1 Problématique

La prévision du trafic offre plusieurs possibilités pour améliorer la performance des réseaux informatiques. Elle permet de déterminer à l'avance l'état du réseau et par suite, les protocoles du réseau et les mécanismes concernés peuvent agir pro-activement sur plusieurs paramètres pour atteindre des objectifs de qualité de service. La prévision peut être appliquée au niveau de la planification, le dimensionnement des réseaux,

1. Un réseau fédérateur est la partie centrale sur laquelle repose un réseau de télécommunication, caractérisée par son haut débit, qui permet d'interconnecter des réseaux plus petits, à l'intérieur d'une entreprise, d'une région ou d'un vaste territoire.

l'amélioration des mécanismes de contrôle de congestion et d'allocation de ressources etc.

Afin de tirer profit de la prévision et de l'exploiter efficacement, il faut obtenir des prévisions suffisamment précises. Pour atteindre cet objectif, il faut d'abord analyser les traces du trafic Internet et étudier leurs caractéristiques. Il faut ensuite étudier les paramètres qui ont une influence sur la précision de la prévision.

Le trafic du réseau Internet est en général un processus complexe et très variable avec le temps. Par conséquent, avant d'effectuer la prévision du trafic, il est nécessaire d'étudier les caractéristiques du trafic et ses propriétés ainsi que les paramètres qui peuvent l'affecter. Certes, les méthodes statistiques standards constituent un outil important permettant de traiter, étudier et modéliser les données représentant le trafic Internet. Ces méthodes englobent principalement des techniques de prétraitement, d'analyse et de modélisation de données.

Ainsi, il faut étudier l'effet des techniques de prétraitement, notamment l'échantillonnage, sur les données représentant le trafic. Ces techniques peuvent mettre en valeur ou cacher certaines caractéristiques. Il est donc important de définir les caractéristiques pertinentes pour la prévision du trafic et proposer une technique capable de les préserver et améliorer la précision de la prévision.

Il est aussi nécessaire d'évaluer si les différents modèles statistiques disponibles sont capables de capturer les caractéristiques du trafic Internet afin d'évaluer leur capacité de représenter le trafic et de le prévoir. Le choix du modèle le plus appropriée est l'une des conditions essentielles pour obtenir une prévision suffisamment précise. De plus, celui-ci a des paramètres qui doivent être choisis ou estimés rigoureusement tout en assurant un bon compromis entre la performance et la complexité. Par conséquent, il faut étudier l'effet de ces différents paramètres à considérer (les données nécessaires pour définir le modèle, les paramètres d'entrée, l'échelle de temps, etc.)

Si la prévision du trafic devient possible avec l'un de ces modèles, plusieurs ap-

plications sont envisageables au niveau de différents types de réseaux et protocoles. Dans le présent travail, on s'intéresse à la gestion des files d'attente et au contrôle du taux de perte dans les réseaux optiques à commutation de rafales (*Optical Burst Switched Networks* - OBS).

Concernant la gestion des files d'attente, la congestion et l'instabilité de la taille des files sont la cause principale de la détérioration de la performance. En effet, la congestion provoque des pertes de paquets ; l'instabilité de la taille des files d'attente provoque des délais de transmission très variables qui réduisent la performance de plusieurs applications. La prévision du trafic pourrait être exploitée afin de prévoir la congestion de la file à l'avance. Ainsi, un mécanisme qui utilise cette prévision permettrait d'éviter les futures congestions et l'instabilité de la taille de la file d'attente.

D'autre part, au niveau des réseaux optiques à commutation de rafales, la garantie de la qualité de service est un objectif crucial pour les fournisseurs de service Internet. En effet, le taux de perte de rafales dans le réseau est l'un des paramètres de qualité de service les plus importants mais qui n'est pas contrôlé par l'administrateur. Un taux de perte fixe dans le réseau permet aux applications d'offrir une qualité de service stable et garantie.

La prévision du trafic combinée avec l'automatique pourraient être utilisées pour proposer une solution efficace pour contrôler ce paramètre et par suite, garantir une qualité de service dans le réseau.

Nous présentons, dans ce qui suit, l'objectif et la contribution de cette thèse.

0.2 Contribution de la thèse

L'objectif de cette thèse est d'étudier les techniques de prévision et d'évaluer la performance des modèles de prévision et de les appliquer pour résoudre différentes problématiques réseaux, plus précisément la gestion des files d'attente et le contrôle du taux de perte dans les réseaux optiques à commutation de rafales. Ce travail s'intéresse

à la prévision à court terme (millisecondes - secondes).

Le premier volet du travail consiste à étudier les différents paramètres qui ont une influence sur la qualité de la prévision en termes d'erreur. Nous analysons les caractéristiques du trafic Internet qui doivent être prises en compte pour définir et paramétrer le modèle de prévision. Les méthodes statistiques d'analyse et de prétraitement, notamment les techniques d'échantillonnage, sont aussi exploitées pour augmenter la précision de la prévision. Ainsi, nous proposons une nouvelle technique d'échantillonnage de données qui préserve les caractéristiques du trafic pertinentes à la prévision du trafic.

Le deuxième volet du travail porte sur l'exploitation de la prévision du trafic afin d'améliorer la gestion des files d'attente et de contrôler le taux de perte dans les réseaux OBS. Ainsi, nous proposons un nouveau mécanisme de gestion de files d'attente basé sur la prévision. Nous prouvons qu'il est capable d'améliorer la performance du réseau en termes de la taille de la file d'attente, de délai d'attente et du débit reçu par la destination. Nous nous intéressons aussi à l'application de la prévision pour contrôler le taux de perte dans les réseaux optiques à commutation de rafales. Nous proposons un modèle basé sur la prévision du trafic et les systèmes asservis afin de contrôler le taux de perte dans les réseaux OBS en tenant compte de la qualité de service.

Par conséquent, la contribution consiste à :

1. Décrire l'état de l'art de la prévision du trafic : les travaux dans le domaine ainsi que les caractéristiques du trafic, les différents modèles de prévision et les techniques de prétraitement.
2. Analyser l'effet des paramètres du modèle de prévision sur la performance de la prévision.
3. Proposer une nouvelle technique d'échantillonnage adaptée pour la prévision du trafic.
4. Proposer un nouveau mécanisme de gestion de files d'attente basée sur la prévision.
5. Proposer un modèle se basant sur la modélisation, la prévision du trafic et l'au-

tomatique afin de contrôler le taux de perte dans les réseaux OBS.

Dans ce qui suit, nous détaillons les différents points de la contribution.

0.2.1 Analyse de l'effet des paramètres du modèle sur la performance de la prévision

L'objectif est d'analyser l'effet des différents paramètres du modèle de prévision et des caractéristiques du trafic sur la performance de la prévision en termes d'erreur. Ainsi, par l'intermédiaire d'expérimentations sur des traces de trafic réel provenant de divers liens Ethernet, nous étudions l'effet de quelques paramètres sur la précision de la prévision. Les modèles de prévision considérés sont : le modèle neuroflou α -SNF et les modèles ARMA et ARIMA. Ils sont appliqués pour prévoir le débit du trafic qui traverse un lien (exprimé Mbps). Les paramètres étudiés sont : la quantité de données requise pour identifier le modèle, le nombre des anciennes observations (appelés retards) utilisées comme entrées pour le modèle, la granularité des données (c.-à-d. l'intervalle de temps séparant deux mesures), la variance des données et la distribution de la taille des paquets.

Les résultats expérimentaux prouvent que les modèles, identifiés avec une faible quantité de données et en utilisant seulement un retard comme entrée, donnent les prévisions les précises.

Nous étudions également l'utilisation des variables exogènes comme entrées pour le modèle. Les variables exogènes sont des variables, qui sont différentes des retards, telles que le nombre de paquets ou le trafic filtré. Nous prouvons que le nombre de paquets, particulièrement le nombre de grands paquets (de tailles supérieures à 800 octets) peut être utilisé pour prévoir efficacement le débit.

0.2.2 Proposition d'une technique d'échantillonnage adaptée à la prévision du trafic

Nous nous proposons d'analyser l'effet des techniques d'échantillonnage de données sur la performance de la prévision. Nous proposons une nouvelle technique qu'on a baptisée *échantillonnage basé sur le maximum* (*maximum-based sampling* - MBS).

Nous montrons analytiquement qu'elle préserve l'auto-similarité et la dépendance à long terme du trafic. Ensuite, nous la comparons aux autres techniques d'échantillonnage en termes de moyenne, variance, sporadicité et auto-similarité. Les résultats montrent que, contrairement aux autres techniques, la technique MBS préserve l'auto-similarité et la dépendance à long terme.

De plus, nous effectuons des prévisions en utilisant les données échantillonnées. Les résultats montrent que les données obtenues avec la technique MBS améliorent la performance du modèle de prévision.

0.2.3 Proposition d'un nouveau mécanisme de gestion de files d'attente basée sur la prévision

Les mécanismes de gestion de files d'attente (*Active Queue Management* - AQM) permettent d'éviter la congestion. Ils se basent sur le rejet proactif (à l'avance) des paquets pour prévenir les sources de la congestion du réseau. Le mécanisme *Random Early Detection* (RED) est l'une des solutions les plus connues pour résoudre le problème de congestion des files d'attente (Floyd et Jacobson, 1993). Cependant, il est incapable de stabiliser la taille de la file et de réaliser une utilisation importante du lien. En effet, il dépend des paramètres définis par l'administrateur du réseau ce qui le rend moins flexible aux variations du trafic. Par conséquent, le résultat n'est pas stable.

Afin de pallier aux défauts du RED, nous proposons α -SNFAQM, un nouveau mécanisme de gestion des files d'attente basé sur la prévision neurofloue du trafic et qui est capable de stabiliser la taille de la file d'attente tout en conservant une utilisation

importante du lien. À l'aide du modèle de prévision α -SNF, le mécanisme proposé est capable de capturer les variations du trafic et prévoir les futures congestions. Ainsi, la technique permet de distinguer deux niveaux de congestion : congestion grave et la congestion légère.

Nous effectuons aussi une étude analytique pour justifier les décisions du mécanisme α -SNFAQM et estimer la taille moyenne de la file (en considérant seulement le trafic TCP et en supposant que les temps d'aller-retour et les taux de perte sont identiques pour toutes les connexions TCP)

Le mécanisme α -SNFAQM a été implémenté dans un module en C++ qui a été intégré au simulateur *Network Simulator 2* (NS-2). Ainsi, nous comparons la performance du mécanisme α -SNFAQM par rapport aux autres mécanismes AQM tels que RED, PAQM (Gao, He et Hou, 2002) et APACE (Jain, Karandikar et Verma, 2003; Jain, Karandikar et Verma, 2004). Les simulations, effectuées sur différentes topologies, montrent que le mécanisme α -SNFAQM offre une meilleure performance par rapport aux autres mécanismes. En effet, il permet de stabiliser la taille de la file d'attente tout en gardant une utilisation élevée du lien. Ceci permet de contrôler les délais d'attente des paquets dans les files d'attente et par suite, contrôler les délais de bout-en-bout. Le mécanisme α -SNFAQM permet aussi de maximiser le débit à la réception (au niveau de la couche application) en minimisant les retransmissions TCP.

0.2.4 Application de la prévision du trafic dans le cas des réseaux OBS

Les réseaux optiques à commutation de rafales (OBS) constituent une nouvelle génération de réseaux optiques.

Nous appliquons la modélisation et la prévision du trafic combinées à l'automatique afin de proposer un système de contrôle asservi avec feedback capable de contrôler le taux de perte dans les réseaux OBS en tenant compte de la qualité de service. Nous proposons ensuite un modèle amélioré avec une technique qui se base sur la prévision et qui permet d'éviter les feedbacks provenant du réseau. Cette amélioration est com-

posée de deux étapes. La première étape consiste à prévoir le taux de perte au niveau de la couche TCP. La deuxième étape est la déduction du taux de perte au niveau de la couche OBS afin de l'utiliser à la place du feedback du réseau.

Nous avons implémenté le modèle proposé dans un module en C++ qui a été intégré au simulateur *Network Simulator 2* (NS-2). Les simulations de différents scénarios montrent que le modèle proposé permet de contrôler efficacement le taux de perte dans le réseau et de le stabiliser autour de la valeur fixée par l'administrateur. Cela est validé dans le cas de réseaux avec ou sans qualité de service. Le modèle amélioré, qui utilise la prévision des pertes au niveau TCP, est capable de contrôler le taux de perte sans aucun feedback du réseau.

0.3 Organisation de la thèse

Le présent travail sera organisé comme suit.

Le premier chapitre décrit l'état de l'art de la modélisation et la prévision du trafic. Nous présentons les techniques de prétraitement des données (notamment les techniques d'échantillonnage), les caractéristiques du trafic Internet et les différents modèles de prévision. Nous présentons aussi les travaux sur la prévision du trafic et ses applications et un rappel sur les réseaux optiques à commutation de rafales et les différents problèmes reliés à la contention et la perte des paquets.

Dans le deuxième chapitre, nous analysons de traces réelles de trafic. Nous étudions aussi le choix des différents paramètres du modèle de prévision et leur influence sur la performance de la prévision en termes d'erreur.

Dans le troisième chapitre, nous proposons une nouvelle technique d'échantillonnage adaptée à la prévision du trafic et qui est capable de préserver l'auto-similarité du trafic.

Dans le quatrième chapitre, nous proposons un nouveau mécanisme de gestion des files d'attente qui utilise la prévision du trafic.

Dans le cinquième chapitre, nous proposons un modèle qui se base sur la prévision du trafic et les systèmes asservis afin de contrôler le taux de perte dans les réseaux OBS.

Finalement, nous présentons les conclusions de ce travail ainsi que les possibles perspectives de recherches.

CHAPITRE I

ÉTAT DE L'ART

«Savoir pour prévoir, afin de pouvoir»

Auguste Comte

Depuis les années quatre-vingt-dix, plusieurs travaux ont été réalisés sur la prévision du trafic afin d'éviter la congestion et la saturation du réseau à court ou à long terme. Plusieurs paramètres ont été analysés et plusieurs modèles ont été évalués pour du trafic provenant de différents types de réseaux.

Pour étudier la prévision du trafic, il est primordial de tenir compte des caractéristiques du réseau et du trafic et des propriétés des différents modèles de prévision.

Ainsi, les objectifs de ce chapitre se résument à :

- Présenter le modèle de la série temporelle qui est utilisé pour modéliser les différentes variables mesurées à partir du réseau.
- Présenter les techniques de prétraitement des données. Elles permettent de supprimer les bruits et les petites variations ou de mettre en valeur certaines caractéristiques. Les données obtenues à la suite de ces prétraitements permettent, généralement, d'estimer avec plus de précision les paramètres des modèles de prévision ; cela améliore la précision de la prévision.

- Présenter les techniques d'analyse du trafic notamment quand la variable étudiée est modélisée par une série temporelle.
- Présenter les principales caractéristiques du trafic Internet telles que son auto-similarité et sa distribution à décroissance lente.
- Présenter les principaux modèles de prévision du trafic et présenter leurs points forts et leurs défauts.
- Présenter les travaux, les principaux résultats et les difficultés existantes dans le domaine de la prévision du trafic, de l'échantillonnage des données ainsi que les mécanismes de gestion des files d'attente, notamment ceux qui utilisent la prévision du trafic.
- Présenter les réseaux optiques à commutation de rafales et les différents problèmes reliés à la contention et la perte des paquets.

1.1 Modélisation du trafic Internet

Afin de caractériser le trafic Internet, plusieurs variables peuvent être étudiées telles que le débit au sein d'un lien (c.-à-d. la quantité de données qui traverse un lien entre deux équipements par intervalle de temps), le temps d'aller-retour¹, le taux de perte des paquets etc. Ces variables sont généralement mesurées directement au niveau des équipements ou calculées à partir des traces de paquets collectées grâce à des cartes réseaux dédiées.

Puisque l'objectif est d'examiner les changements du trafic Internet au cours du

1. Le temps d'aller-retour (*Round Trip Time* - RTT) est le temps entre l'envoi d'un paquet et la réception de la réponse.

temps, les variables étudiées sont généralement modélisées par des séries temporelles.

Une *série temporelle* est définie comme étant une suite d'observations d'une variable y ordonnées dans le temps. Ainsi, on note par $y(t)$ la valeur de y à la date d'observation t . Une série temporelle $y = \{y(t); t \in \mathbb{N}\}$ peut être vue comme étant une réalisation d'un processus stochastique discret $Y = \{Y(t); t \in \mathbb{N}\}$.

Un rappel sur les processus stochastiques et les variables aléatoires est présenté dans l'appendice A.

Le présent travail s'intéresse à la modélisation et la prévision du trafic et plus précisément des séries temporelles représentant les différentes variables qui le caractérisent. La variable la plus importante est le débit des données qui traversent un lien du réseau, elle est souvent exprimée en Mbps ou en *paquets par seconde* (pps).

Dans ce qui suit, nous présentons les techniques de prétraitement des données collectées (les valeurs de la série temporelle) avant leur utilisation pour l'analyse et l'identification des paramètres des modèles de prévision.

1.2 Les techniques de prétraitement des données

Plusieurs techniques sont disponibles pour traiter les données obtenues à partir des mesures du trafic. Les objectifs du prétraitement des données dépendent de l'utilisation ultérieure de celles-ci. Parmi ces objectifs, on note la suppression des petites variations, la normalisation des données sur une même échelle, le filtrage du bruit et la mise en évidence de certaines caractéristiques des données. Par conséquent, ces techniques facilitent l'analyse du trafic et l'identification du modèle de prévision, ce qui améliore la performance du modèle en termes d'erreur.

Dans ce qui suit, nous présentons les techniques les plus fréquentes de prétraitement du trafic.

1.2.1 Les méthodes de lissage

Les méthodes de lissage visent à capturer les caractéristiques importantes des données tout en supprimant les bruits ou les petites variations.

La méthode de *la moyenne mobile* (appelée aussi « *filtre* ») est l'une des méthodes de lissage les plus courantes (Gourieroux et Monfort, 1990). Son principe consiste à pondérer les observations par des poids. La moyenne mobile d'ordre k , notée $y'(t)$, est donnée par :

$$y'(t) = \sum_{i=0}^{k-1} a_i y(t-i), \quad (1.1)$$

où a_i est le poids associé à $y(t-i)$ et on a $\sum_{i=0}^{k-1} a_i = 1$.

Cette formulation générale laisse les valeurs des pondérations (a_i) indéterminées. D'un point de vue pratique, on choisit une valeur égale à $1/k$ pour tous les poids a_i .

L'inconvénient de l'équation 1.1 est le décalage introduit vu que l'on n'utilise que les valeurs passées et présentes. Pour y remédier, on utilise *une moyenne mobile centrée* donnée par :

$$y'(t) = \frac{1}{2k+1} \sum_{i=-k}^{i=k} y(t+i). \quad (1.2)$$

Le lissage peut être utilisé lors du passage d'une échelle de temps (appelée *granularité*) vers une échelle plus grande. Dans ce cas, les petites variations des données vont disparaître lorsqu'on augmente l'échelle.

D'autre part, si, dans l'équation 1.1, tous les poids a_i sont égaux à un, on ne parle plus de lissage, mais plutôt d'*agrégation de trafic*.

Les méthodes de lissage sont importantes dans le cadre de la prévision. En effet, les données servent généralement à estimer les paramètres du modèle de prévision. Ainsi, la réduction des bruits et des petites variations des données permet d'améliorer la précision de ces paramètres et, par suite, la précision du modèle.

1.2.2 Traitement logarithmique

Le principe du traitement logarithmique consiste à associer à toute valeur de la série temporelle $y(t)$ une valeur $y'(t)$ telle que (Rosenfeld, Hannoset et Sabatier, 1988)

$$y'(t) = \log(y(t)). \quad (1.3)$$

Le premier avantage de ce traitement est que l'écart entre $y'(t_1)$ et $y'(t_2)$ est en fonction du rapport entre $y(t_1)$ et $y(t_2)$. En effet, on a :

$$\begin{aligned} y'(t_2) - y'(t_1) &= \log(y(t_2)) - \log(y(t_1)) \\ &= \log\left(\frac{y(t_2)}{y(t_1)}\right) \\ &= \log\left(\frac{y(t_1) + (y(t_2) - y(t_1))}{y(t_1)}\right) \\ &= \log\left(1 + \frac{y(t_2) - y(t_1)}{y(t_1)}\right), \end{aligned} \quad (1.4)$$

donc $y'(t_2) - y'(t_1) = \log(1 + r)$ où $r = \frac{y(t_2) - y(t_1)}{y(t_1)}$ est la variation relative entre $y(t_1)$ et $y(t_2)$. En d'autres termes, les variations relatives de $y(t)$ correspondent aux déplacements verticaux en ordonnées logarithmiques $y'(t)$. Par exemple, lorsque $y(t)$ varie de 100 à 140 ou de 1000 à 1400, la variation relative est toujours égale à 40 % et les déplacements verticaux en ordonnées logarithmiques sont de mêmes longueurs.

Le deuxième avantage de ce traitement est que, lorsqu'une fonction est multipliée par un coefficient constant, sa courbe représentative subit une simple translation verticale. En effet, on a :

$$\log(ky) = \log(k) + \log(y) = K + y', \quad (1.5)$$

où $K = \log(k)$ représente l'amplitude de la translation.

On procède à une transformation logarithmique si la série croît d'une manière exponentielle. Par exemple, les valeurs de la charge d'un réseau fédérateur (exprimées en nombre de paquets ou en Mbps) augmentent d'une manière exponentielle ; la transformation logarithmique permet de remettre toutes les valeurs sur une même échelle

(Groschwitz et Polyzos, 1994). Une fois que la prévision est effectuée, l'opération inverse permet de retrouver l'échelle originale des données.

Dans le cadre de ce travail, nous étudions la prévision à court terme : la croissance exponentielle du trafic n'est pas visible sur des intervalles de temps restreints (allant de quelques millisecondes jusqu'aux heures). Par conséquent, nous n'aurons pas recours à la transformation logarithmique.

1.2.3 Les techniques d'échantillonnage

L'échantillonnage d'une série temporelle consiste à sélectionner un certain nombre de ses éléments, appelés *échantillons*. Plusieurs techniques d'échantillonnage ont été proposées par des chercheurs (Cozzani et Giordano, 1998; Feldmann et al., 2001; Eitan et Varghese, 2002; Duffield, Lund et Thorup, 2005; Duffield et Grossglauser, 2000; He et Hou, 2006) et par les groupes de travail IETF IPFIX (Internet Protocol Flow Information eXport (IPFIX),) et PSAMP (Packet Sampling (PSAMP) - IETF Working Group, 2009).

Soit $\{y'(t); t \in \mathbb{N}\}$ la série échantillonnée à partir de la série $\{y(t); t \in \mathbb{N}\}$. Il y a trois techniques d'échantillonnage qui sont généralement utilisées pour le trafic Internet (fig. 1.1) :

- Échantillonnage aléatoire : les éléments de $\{y'(t)\}$ sont sélectionnés aléatoirement (c.-à-d. avec probabilité égale) à partir de la série $\{y(t)\}$. Ainsi, on a :

$$\begin{cases} y'(0) = y(i_0), & i_0 \text{ est choisi aléatoirement tel que } i_0 \geq 0 \\ \text{si } y'(t) = y(i) \text{ alors } y'(t+1) = y(j), & j \text{ est choisi aléatoirement tel que } j \geq i. \end{cases}$$

- Échantillonnage systématique : les éléments qui sont mesurés à des instants t multiples d'un entier $c > 0$ sont automatiquement sélectionnés. Par suite, on a :

$$y'(t) = y(ct), \quad \forall t \geq 0.$$

Le choix du paramètre c dépend de l'utilisation ultérieure de la série échantillonnée.

- Échantillonnage aléatoire stratifié : l'axe du temps est divisé en intervalles de longueur c (appelés *strates*). La longueur c est mesurée en nombre de valeurs de $y(t)$ à considérer dans l'intervalle en question. Cette technique consiste à choisir aléatoirement un élément $y'(t)$ à partir de chaque intervalle (avec la même probabilité pour tous les éléments $y(t)$ de cet intervalle). Ainsi, pour $t \geq 0$, on a :

$$y'(t) = y(i), \text{ } i \text{ est choisi aléatoirement tel que } i \in [tc, (t+1)c[.$$

Le choix de la longueur de l'intervalle c dépend de l'utilisation ultérieure de la série échantillonnée.

L'inconvénient de l'échantillonnage aléatoire est qu'il ne conserve pas une distance égale entre les échantillons en termes de temps. Par exemple, si $y(t)$ représente le débit mesuré chaque 0,2 seconde, on peut faire un échantillonnage aléatoire systématique ou aléatoire stratifié avec $c = 5$ pour obtenir une valeur de $y'(t)$ qui sera un échantillon représentatif pour l'intervalle d'une seconde ($5 \times 0,2$ s). Ceci n'est pas possible avec l'échantillonnage aléatoire puisque la taille de l'intervalle de sélection des éléments n'est pas fixe ou bornée.

Le choix d'une technique d'échantillonnage doit tenir compte de divers critères qui dépendent principalement de l'utilisation ultérieure des données. Par exemple, si les données seront utilisées pour la prévision du trafic, il serait intéressant que les données échantillonnées gardent la forte auto-corrélation, l'auto-similarité et la haute variabilité du trafic. Dans le chapitre 3, nous proposons une technique d'échantillonnage, qui génère des données permettant d'améliorer la performance de la prévision en termes d'erreur.

Une fois que le prétraitement des données est effectué, les valeurs de la série temporelle obtenue sont analysées afin de repérer les différentes propriétés et caractéristiques du trafic.

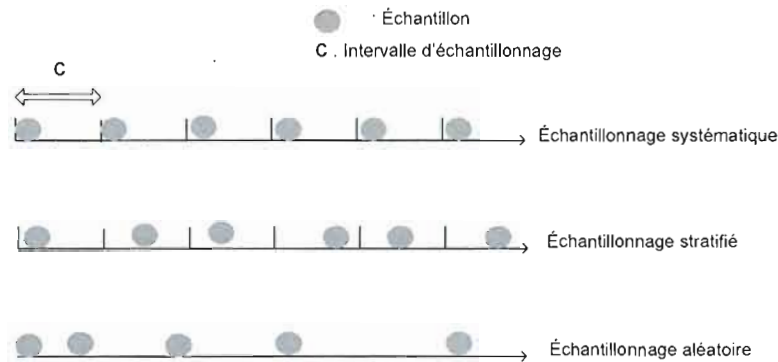


Figure 1.1 Illustration des trois techniques d'échantillonnage (He et Hou, 2005).

1.3 Analyse du trafic Internet

L'analyse du trafic Internet consiste à analyser les différentes variables qui peuvent être mesurées à partir du réseau. Ces variables, notamment le débit au sein d'un lien, sont modélisées par des séries temporelles. Ainsi, il est primordial de présenter les différentes approches pour l'analyse des séries temporelles.

Une première approche est constituée par l'analyse spectrale directement importée du domaine de la physique. Elle se base principalement sur la représentation des données dans le domaine fréquentiel afin de rechercher les périodicités.

Une seconde approche est l'analyse des dépendances temporelles qui consiste à étudier directement les corrélations entre la valeur actuelle de la série et ses valeurs passées.

1.3.1 Analyse spectrale (recherche des périodicités)

Généralement, on connaît la périodicité d'une série représentant le trafic Internet. Celle-ci s'appuie le plus souvent sur des cycles connus : mois, jour et trimestre. Cependant, dans certains cas, les cycles ne sont pas aisément identifiables. De plus, ils

peuvent ne pas correspondre à une périodicité connue.

L'analyse spectrale permet d'identifier ces périodicités. Ce type d'analyse s'appuie généralement sur la transformée de Fourier qui fournit une bonne description de la série. Cependant, la transformée de Fourier a de nombreuses limitations quand la série temporelle à analyser n'est pas stationnaire (Lin, 1976).

1.3.2 Analyse des dépendances temporelles

Une série temporelle ne dépend pas seulement des facteurs externes, mais aussi de ses états passés. Dans ce qui suit, nous présentons les fonctions d'autocorrélation qui permettent d'analyser cette dépendance.

– Fonction d'autocorrélation (ACF) :

Définition 1.1. Soit $r(k)$ l'autocorrélation d'ordre k de la série temporelle $y(t)$. Elle est obtenue en divisant la covariance des séries $y(t)$ et $y(t - k)$ par la variance de la série $y(t)$, c.-à-d. :

$$r(k) = \frac{\sum_{t=1}^{n-k} (y(t) - \bar{y})(y(t+k) - \bar{y})}{\sum_{t=1}^n (y(t) - \bar{y})^2} \quad (1.6)$$

où $y(t)$ est la valeur de la série temporelle à l'instant t , n est la taille des données disponibles (c.-à-d. le nombre de valeurs de $y(t)$ disponibles) et \bar{y} est la moyenne estimée sur l'ensemble des valeurs $y(t)$ de la série.

L'autocorrélation d'ordre k d'une série temporelle mesure la corrélation entre cette série et elle-même avec un retard de k mesures. Le graphe de la fonction d'autocorrélation (c.-à-d. $r(k)$ en fonction de k) est appelé *corrélogramme* (ACF).

– Fonction d'autocorrélation partielle (PACF) :

Définition 1.2. Soit $r^*(k)$ la corrélation partielle d'ordre k d'une série temporelle $y(t)$. Elle est calculée selon les formules suivantes :

$$r^*(1) = r(1) ,$$

$$r^*(2) = \frac{r(2) - r^2(1)}{1 - r^2(1)},$$

$$r^*(k) = \frac{r(k) - \sum_{j=1}^{k-1} \phi_{k-1,j} r(k-j)}{1 - \sum_{j=1}^{k-1} \phi_{k-1,j} r(j)}, \quad (1.7)$$

avec les équations de récurrences suivantes :
$$\begin{cases} \phi_{k,j} = \phi_{k-1,j} - \phi_{k,k} \phi_{k-1,k-j} \\ \phi_{k,k} = r^*(k) \end{cases}$$

où $r(k)$ est l'autocorrélation d'ordre k (éq. 1.6).

La signification des corrélations partielles est simple en dépit d'une forme analytique compliquée : elle mesure la corrélation entre la série $y(t)$ et son retard $y(t-k)$ d'ordre k lorsque l'influence des variables $y(t-k-i)$, $k < i$ a été retirée. Le graphe de la fonction d'autocorrélation partielle (c.-à-d. $r^*(k)$ en fonction de k) est appelé *corrélogramme partiel* (PACF).

Les études de la décennie passée sur le trafic de réseau ont montré d'une façon convaincante que le trafic des réseaux locaux aussi bien que celui des réseaux étendus sont caractérisés par une forte autocorrélation et une extrême variabilité (Leland et al., 1994; Willinger, Paxson et Taquq, 1998; Park et Willinger, 2000). Dans ce qui suit, nous présentons les principales caractéristiques du trafic Internet.

1.4 Caractéristiques du trafic Internet

L'importante autocorrélation et l'extrême variabilité du trafic Internet sont la conséquence de sa non-stationnarité ainsi que son caractère *d'auto-similarité* et de *dépendance à long terme* (*long range dependence* - LRD) (Leland et al., 1994; Willinger, Paxson et Taquq, 1998; Park et Willinger, 2000). Dans ce qui suit, nous présentons ces caractéristiques ainsi que leurs formulations mathématiques.

1.4.1 Non-stationnarité

Définition 1.3. *Un processus $\{Y(t); t \geq 0\}$ est dit stationnaire au sens large si son espérance² et ses autocovariances sont invariantes par translation dans le temps :*

- $\forall t \in \mathbb{N}, \mathbb{E}[Y(t)] = \mu$ où μ est une constante,
- $\forall t, h \in \mathbb{N}, \text{Cov}(Y(t), Y(t+h)) = \text{Cov}(Y(0), Y(h))$.

Ainsi, l'autocovariance entre $Y(t)$ et $Y(t+h)$ ne dépend que de l'intervalle h et non de la date t . Par conséquent, la variance est constante pour un processus stationnaire :

$$\text{Var}[Y(t)] = \text{Cov}(Y(0), Y(0)) \quad \forall t \in \mathbb{N}.$$

Définition 1.4. *Un processus $\{Y(t); t \geq 0\}$ est dit stationnaire au sens strict si*

$$\forall h, t_1, \dots, t_n,$$

$$f_{Y(t_1), \dots, Y(t_n)} = f_{Y(t_1+h), \dots, Y(t_n+h)}, \quad (1.8)$$

où $f_{Y(t_1), \dots, Y(t_n)}$ est la fonction de distribution conjointe de $Y(t_1), \dots, Y(t_n)$

(app. A, déf. A.6).

Ainsi, un processus est stationnaire lorsque sa structure probabiliste (sa distribution) est stable au cours du temps (c.-à-d. indépendante du temps). Une série temporelle $y(t)$ est stationnaire si elle est la réalisation d'un processus stationnaire.

Les études effectuées sur le trafic réel collecté sur Internet montrent que le trafic n'est pas stationnaire. En effet, le trafic se caractérise par une périodicité mensuelle, hebdomadaire et journalière (Papagiannaki et al., 2003b; Paxson et Floyd, 1995; Groschitz et Polyzos, 1994; Zhani, 2005). Cependant, des travaux ont montré qu'il est possible de considérer que le trafic est stationnaire au sens large seulement à très faible échelle de temps (des millisecondes jusqu'aux minutes) (Zhang et Duffield, 2001).

La non-stationnarité rend la prévision une tâche difficile puisque les caractéristiques statistiques du trafic sont très variables au cours du temps. Cependant, il existe

2. L'espérance $\mathbb{E}[Y(t)]$, la variance $\text{Var}[Y(t)]$ et l'autocovariance $\text{Cov}(Y(s), Y(t))$ sont définies dans l'appendice A.

des techniques permettant de stationnariser des données. La méthode la plus utilisée pour stationnariser les données est appelée *la différenciation*.

La différenciation d'ordre d de la série $y(t)$ est la série $(1 - L)^d y(t)$ où L est l'opérateur « retard » (app. A, déf. A.15).

Nous pouvons ainsi effectuer plusieurs différenciations d'une façon répétée (d fois) jusqu'à ce que la série devienne stationnaire (Box et Jenkins, 1976).

1.4.2 Auto-similarité et dépendance à long terme

La présence de l'auto-similarité et de la dépendance à long terme dans le trafic est traduite par des corrélations importantes avec le passé sur différentes échelles de temps. De plus, un trafic auto-similaire garde la même structure des variations quelle que soit l'échelle de temps (l'échelle d'observation) considérée. Cette propriété est en opposition au processus de Poisson (app. A, déf. A.14), un processus à court terme, où les variations s'annulent lorsque l'échelle de temps augmente (Leland et al., 1994).

Dans ce qui suit, nous présentons les formulation mathématiques des processus auto-similaires et de la dépendance à long terme (Willinger, Paxson et Taqqu, 1998; Leland et al., 1994).

- Auto-similarité :

Définition 1.5. *Un processus continu $Y = \{Y(t); t \geq 0\}$ est dit auto-similaire d'ordre H s'il satisfait à la condition suivante :*

$$Y(t) =_d a^{-H} Y(at), \quad \forall t \geq 0, \quad \forall a > 0, \quad \frac{1}{2} \leq H < 1 \quad (1.9)$$

où H est appelé le paramètre de Hurst et le symbole $=_d$ signifie une égalité en distribution (app. A, déf. A.5).

Il existe aussi une seconde définition de l'auto-similarité en temps discret qui est plus appropriée à la théorie standard des séries temporelles (Willinger, Paxson et Taqqu, 1998).

Définition 1.6. Soit $X = \{X(t); t \in \mathbb{N}\}$ un processus discret et stationnaire, le processus agrégé d'ordre m , noté $X^m = \{X^m(t); t \in \mathbb{N}\}$, est défini comme étant le processus obtenu par le calcul de la moyenne des éléments du processus original X contenus dans des blocs contigus de taille m :

$$X^m(k) = \frac{1}{m} \sum_{t=(k-1)m+1}^{km} X(t). \quad (1.10)$$

Définition 1.7. Un processus discret X est dit auto-similaire d'ordre H si quel que soit l'entier $m > 0$, X^m a la même distribution que X multipliée par m^{H-1} , c.-à-d. :

$$X(t) =_d m^{1-H} X^m(t) \quad (1.11)$$

où H est le paramètre de Hurst ($\frac{1}{2} \leq H < 1$) et le symbole $=_d$ signifie une égalité en distribution (app. A, déf. A.5).

Théorème 1.1. Soit $Y = \{Y(t); t \geq 0\}$ un processus continu auto-similaire d'ordre H . Si le processus discret $X = \{X(t) = Y(t) - Y(t-1); t \in \mathbb{N}\}$ est stationnaire alors il est autosimilaire d'ordre H (Park et Willinger, 2000).

Le processus continu Y est généralement interprété comme étant le volume de données cumulatif qui traversent le lien (exprimé en Mb ou en paquets). Ainsi, le processus discret X peut être interprété comme étant le débit à l'intervalle $[t, t+1[$, c'est-à-dire la quantité de données qui a traversé le lien pendant cet intervalle (exprimée en Mbps ou en paquets par seconde).

D'autre part, la fonction d'autocorrélation du processus X , notée $r(k)$, vérifie pour $\frac{1}{2} < H < 1$ (Park et Willinger, 2000) :

$$r(k) =_d k^{2H-2}, \quad k \rightarrow \infty \quad (1.12)$$

où H est le même paramètre de Hurst.

L'une des caractéristiques du trafic auto-similaire est que le degré d'auto-similarité peut être mesuré en utilisant seulement le paramètre de Hurst (Crovella et Bestavros,

1997). Plusieurs algorithmes ont été proposés afin d'estimer le paramètre de *Hurst* à partir d'un ensemble de données (Abry et Veitch, 1998).

- **Dépendance à long terme :**

La dépendance à long terme est l'une des propriétés les plus importantes des processus auto-similaires. Elle est définie par le comportement de l'autocorrélation du processus.

Définition 1.8. *Un processus présente une dépendance à long terme si sa fonction d'auto-corrélation, notée $r(k)$, décroît hyperboliquement avec k et ne s'annule pas. Ceci est traduit mathématiquement par (Park et Willinger, 2000) :*

$$\sum_{k=-\infty}^{\infty} r(k) = \infty. \quad (1.13)$$

Si la somme $\sum_{k=-\infty}^{\infty} r(k)$ est finie alors le processus présente une dépendance à court terme.

Ainsi, en examinant l'équation 1.12, la valeur du paramètre H permet d'identifier si un processus présente une dépendance à long terme (Park et Willinger, 2000) :

- Si $0 < H \leq \frac{1}{2}$, on a $\sum_{k=-\infty}^{\infty} r(k) = 0$: le processus n'est pas auto-similaire et il ne présente pas une dépendance à long terme, mais, plutôt une dépendance à court terme (Abry et Veitch, 1998).
- Si $\frac{1}{2} < H < 1$, on a $\sum_{k=-\infty}^{+\infty} r(k) = \infty$: le processus est auto-similaire avec une dépendance à long terme. Plus le paramètre H s'approche de 1, plus le degré d'auto-similarité et de dépendance à long terme augmente.

Le trafic auto-similaire est responsable de beaucoup d'effets indésirables dans le réseau, notamment l'importante variance du débit au sein du réseau. Ceci engendre des tailles de files d'attente très variables (Park, Kim et Crovella, 1997; Leland et al., 1994). Par conséquent, les temps d'attente au niveau des routeurs deviennent très variables.

Par contre, l'importante auto-corrélation, qui est actuellement présente dans le

trafic, peut être judicieusement exploitée pour prévoir le trafic avec précision. Plusieurs travaux recommandent d'utiliser des modélisations du trafic qui tiennent compte de l'auto-similarité et de la dépendance à long terme (Leland et al., 1994; Paxson et Floyd, 1995; Park et Willinger, 2000; Willinger, Paxson et Taqqu, 1998; Owezarski et Larrieu, 2004; Park, Kim et Crovella, 1997; Abry et Veitch, 1998).

1.4.3 Distribution à décroissance lente

Il s'est avéré que plusieurs variables caractérisant le trafic Internet ont des distributions à décroissance lente (*Heavy-tailed distribution*) (Leland et al., 1994; Park, Kim et Crovella, 1997; Charzinski, 2000; Downey, 2001). Parmi ces variables, on cite la taille des fichiers transmis et la durée des connexions. Intuitivement, une distribution à décroissance lente attribue des probabilités relativement élevées à des régions éloignées de la moyenne ou de la médiane. Ainsi, elle présente une variance très importante. Ceci montre que la distribution à décroissance lente est à l'origine de l'auto-similarité et de la dépendance à long terme du trafic dans le réseau (Charzinski, 2000).

Définition 1.9. Une variable aléatoire Z est à décroissance lente si :

$$F(x) = P[Z > x] \sim cx^{-\alpha}, x \rightarrow \infty \quad (1.14)$$

où α est appelé l'index de la distribution tel que $0 < \alpha < 2$ et c est une constante positive. Lorsque $1 < \alpha < 2$, la distribution à décroissance lente est caractérisée par une variance infinie et une moyenne finie. Lorsque $0 < \alpha \leq 1$, la moyenne devient aussi infinie (Park et Willinger, 2000).

La distribution de Pareto est une distribution à décroissance lente qui a été souvent utilisée pour représenter avec une bonne précision la distribution de la taille des fichiers et de la durée des connexions et des sessions HTTP³ (Leland et al., 1994; Downey, 2001; Mah, 1997; Nabe et Miyahara, 1998).

3. *Hypertext Transfer Protocol* (HTTP) est un protocole de la couche application destiné à la transmission des pages Web depuis un serveur vers un navigateur client.

Définition 1.10. Une variable aléatoire Z possède une distribution de Pareto si (Park et Willinger, 2000) :

$$P[Z > x] = \left(\frac{x}{x_{min}}\right)^{-\alpha}, x_{min} \leq x \quad (1.15)$$

où α est l'index de la distribution tel que $0 < \alpha < 2$ et x_{min} est une constante.

1.4.4 La sporadicité du trafic

La sporadicité (*burstiness*) du trafic signifie l'existence de périodes concentrées d'activités importantes et de périodes de faibles activités. Ainsi, le trafic se comporte en rafales et crée beaucoup de problèmes au niveau de la gestion du trafic (réservation des ressources) ainsi qu'au niveau des mémoires tampons dans les routeurs.

Dans des travaux précédents, le paramètre de *Hurst* a été considéré comme une métrique pour mesurer la sporadicité du trafic. Cette considération a été justifiée par l'observation visuelle du trafic et par l'hypothèse que le trafic se comporte comme un bruit Gaussien fractionnaire (Leland et Wilson, 1991). En plus du paramètre de *Hurst*, nous allons utiliser un autre critère pour mesurer la sporadicité du trafic échantillonné. Ce critère reflète directement les pics existants dans le trafic. Les critères traditionnellement utilisés sont le coefficient pic sur moyenne, le coefficient de variation ainsi que l'indice de dispersion (Leland et al., 1994). Dans ce travail, nous mesurons la sporadicité avec le ratio *pic par moyenne* (*Peak to Mean* - PM) qui est obtenu divisant le pic de la bande passante par la moyenne de la bande passante. Le PM est donné par la formule suivante :

$$PM = \frac{m}{n} \sum_{k=1}^{n/m} \frac{\max_{i=(k-1)m+1}^{km} (y(i))}{\sum_{j=(k-1)m+1}^{km} \frac{y(j)}{m}}, \quad (1.16)$$

où $y(i)$ est une série temporelle, m le nombre de mesures considérées par intervalle d'étude et n est le nombre d'observations disponibles de la série $y(i)$.

Nous présentons, dans la section suivante, les principaux modèles de prévision utilisés dans les travaux précédents ainsi que la présentation des réseaux optiques à commutation de rafales.

1.5 Les modèles de prévision du trafic

Le but de la prévision est de pouvoir estimer à un moment donné la valeur future d'une variable avec la meilleure précision possible.

Soit la série temporelle $y(t-n), y(t-n+1), \dots, y(t)$, il s'agit de prévoir la valeur de la série temporelle à l'instant $t+h$ c'est-à-dire $y(t+h)$ où h est appelé *horizon prévisionnel*.

Le modèle de prévision peut être représenté par une fonction f qui, à partir des données disponibles c'est-à-dire les anciennes observations $y(t-n), y(t-n+1), \dots, y(t)$ permet de calculer $\hat{y}(t+h)$, la valeur prédite pour $y(t+h)$. Ainsi, on a :

$$\hat{y}(t+h) = f(y(t), y(t-1), \dots, y(t-n)). \quad (1.17)$$

Une ancienne observation $y(t-i)$, $i \geq 1$, est appelée *un retard*.

Il est possible d'avoir recours à des variables additionnelles $x_1(t-i), x_2(t-j), \dots, x_p(t-k)$ afin de prévoir $y(t+h)$. Ces variables sont différentes des anciennes observations de $y(t)$ et sont mesurées à des instants qui précèdent $t+h$. Elles sont appelées des *variables exogènes*. Ainsi, on a :

$$\hat{y}(t+h) = f(x_1(t-i), x_2(t-j), \dots, x_p(t-k)). \quad (1.18)$$

où $(t-i)$, $(t-j)$ et $(t-k)$ représentent les instants de mesure des variables x_i .

Plusieurs modèles pour les séries temporelles ont été exploités dans le cadre de la prévision du trafic. Dans ce qui suit, nous présentons ceux qui sont les plus utilisés, à savoir : le modèle autorégressif à moyenne mobile intégrée, le modèle linéaire des moindres carrés, les modèles auto-similaires, les modèles Markoviens, le filtre de Kalman et les réseaux neuroflous.

1.5.1 Le modèle autorégressif à moyenne mobile (ARMA)

Les modèles linéaires de prévision les plus utilisés sont : le modèle *autorégressif* (*AutoRegressive* - AR), le modèle de la *moyenne mobile* (*Moving Average* - MA) ainsi que leur combinaison : le modèle *autorégressif à moyenne mobile* (*AutoRegressive Moving Average* - ARMA) (Sang et qi Li, 2000; Groschwitz et Polyzos, 1994; Papagiannaki et al., 2003b; Sang et qi Li, 2000; Qiao, Skicewicz et Dinda, 2002; Qiao, Skicewicz et Dinda, 2003; Elbiaze et al., 2009; Sang et qi Li, 2002). Une série temporelle $y(t)$ suit un processus ARMA(p, q) si elle est stationnaire et si pour chaque t :

$$y(t) = \phi_1 y(t-1) + \dots + \phi_p y(t-p) + \epsilon(t) + \theta_1 \epsilon(t-1) + \dots + \theta_q \epsilon(t-q), \quad (1.19)$$

où ϕ_i et θ_j sont des réels et $\epsilon(t)$ est un bruit blanc (app. A, déf. A.8) de variance σ^2 .

L'équation 1.19 peut être écrite d'une façon plus concise :

$$y(t) = \sum_{i=1}^p \phi_i L^i y(t) + (1 + \sum_{i=1}^q \theta_i L^i) \epsilon(t), \quad (1.20)$$

où L est l'opérateur « retard » (app. A, déf. A.15). Ce modèle emploie $p + q + 1$ paramètres inconnus estimés à partir des données disponibles.

Le modèle ARMA combine les modèles AR et MA :

- Le modèle autorégressif AR(p) est le modèle ARMA(p, q) quand $q = 0$. Il contient p paramètres $\phi_1, \phi_2, \dots, \phi_p$. Il est à noter que la fonction d'auto-corrélation partielle (PACF) du processus AR(p) s'annule au-delà de l'ordre p . Sa fonction d'autocorrélation (ACF) tend progressivement vers zéro.
- Le modèle de la moyenne mobile MA(q) est le modèle ARMA(p, q) quand $p = 0$. Il contient $q + 1$ paramètres inconnus $\theta_1, \theta_2, \dots, \theta_q, \sigma^2$ relatifs à la partie MA(q). Il est à noter que la fonction d'autocorrélation (ACF) du processus MA(q) s'annule au-delà de l'ordre q . Sa fonction d'autocorrélation partielle (PACF) tend progressivement vers zéro.

Pour un processus ARMA, les corrélogrammes ACF et PACF tendent progressivement vers zéro. Ainsi, le processus ARMA(p, q) est considéré à dépendance à court terme.

Les corrélogrammes ACF et PACF permettent d'estimer les paramètres p et q . Les paramètres ϕ_i et θ_i peuvent être estimés grâce à la méthode *d'estimation à maximum de vraisemblance* (*Maximum-Likelihood Estimation* - MLE) (Aldrich, 1997).

Si la série temporelle présente des variations qui pourront violer l'hypothèse de stationnarité, la stationnarisation par différenciation à l'ordre d peut être appliquée (sect. 1.4.1). Ainsi, une série temporelle $y(t)$ suit un processus autorégressif à moyenne mobile intégrée (*AutoRegressive Integrated Moving Average* - ARIMA) ARIMA(p, d, q) si $(1 - L)^d y(t)$ suit le processus ARMA(p, q) (d est un entier). Ainsi, on a :

$$(1 - \sum_{i=1}^p \phi_i L^i)(1 - L)^d y(t) = (1 + \sum_{i=1}^q \theta_i L^i) \epsilon(t). \quad (1.21)$$

Ainsi, le modèle ARIMA est valable même pour des données non stationnaires à condition que celles-ci deviennent stationnaires au bout de d différenciations.

1.5.2 Le modèle linéaire des moindres carrés (LMMSE)

Le modèle linéaire des moindres carrés (*Linear Minimum Mean Square Error* - LMMSE) consiste à prévoir la prochaine valeur $y(t+1)$ de la série $y(t)$ comme la somme pondérée des n dernières observations (He et al., 2002; Gao, He et Hou, 2002; He et Hou, 2005; Elbiaze et al., 2009). Ainsi, $y(t+1)$ est donnée par :

$$y(t+1) = [a_1 \ a_2 \ \dots \ a_n] \begin{bmatrix} y(t) \\ \vdots \\ y(t-n+1) \\ y(t-n) \end{bmatrix}, \quad (1.22)$$

où a_1, a_2, \dots, a_n sont les coefficients du modèle LMMSE.

Ces coefficients sont donnés par :

$$[a_1 \ a_2 \ \cdots \ a_n] = [R(1) \ \cdots \ R(n-1) \ R(n)] \times \begin{bmatrix} R(0) & R(1) & \cdots & R(n-1) \\ R(1) & R(0) & \cdots & R(n-2) \\ \cdots & \cdots & \cdots & \cdots \\ R(n-1) & R(n-2) & \cdots & R(0) \end{bmatrix}^{-1} \quad (1.23)$$

où $R(i)$ est la fonction de covariance de la série qui peut être estimée par :

$$R(i) = \frac{1}{n} \sum_{t=i+1}^n y(t)y(t-i), \quad 0 \leq i \leq n-1. \quad (1.24)$$

Il y a deux différences entre le modèle LMMSE et le modèle ARIMA :

- Il n'y a pas de composante de la moyenne mobile (MA) dans le modèle LMMSE qui tiennent compte du bruit blanc $\epsilon(t)$ (éq. 1.19).
- Les paramètres a_i sont calculés instantanément à chaque estimation en fonction des n derniers retards. Pour le modèle ARIMA, les paramètres ϕ_i (éq. 1.19) sont estimés à partir de toutes les traces du trafic disponibles. Ensuite, les mêmes paramètres ϕ_i sont utilisés pour toutes les estimations de $y(t+1)$.

Le choix du nombre de retards n va être étudié davantage dans le chapitre suivant.

1.5.3 Les modèles auto-similaires

Les travaux de recherche ont montré le caractère auto-similaire et la dépendance à long terme du trafic (tel que défini à la section 1.4.2). Ainsi, les modèles traditionnels, tels que le processus de Poisson, les modèles AR, MA ou ARIMA, ne peuvent pas capturer ces dépendances à long terme (Leland et al., 1994; Paxson et Floyd, 1995; Park et Willinger, 2000; Willinger, Paxson et Taqqu, 1998; Owezarski et Larrieu, 2004; Park, Kim et Crovella, 1997; Abry et Veitch, 1998).

Des modèles auto-similaires ont été proposés pour modéliser le trafic. Parmi ces modèles, on cite le *mouvement Brownien fractionnaire* (*Fractional Brownian Motion* - FBM) (Mandelbrot et Ness, 1968) et le modèle *bruit fractionnaire Gaussien* (*Fractional*

Gaussian Noise - FGN). Ces modèles ne peuvent décrire que la dépendance à long terme. D'autres modèles, tels que le modèle ARIMA fractionnaire (FARIMA), sont capables de décrire la dépendance à long et à court terme du trafic.

Le modèle « mouvement Brownien fractionnaire »

Le processus *mouvement Brownien fractionnaire* (*Fractional Brownian Motion* - FBM) a été défini par (Mandelbrot et Ness, 1968).

Un processus $\{X(t); t \geq 0\}$ est un *mouvement Brownien fractionnaire* (FBM) s'il est un processus Gaussien (app. A, déf. A.9) d'espérance nulle et vérifiant :

$$\text{Cov}(X(s), X(t)) = \frac{1}{2}(t^{2H} + s^{2H} - |t - s|^{2H}),$$

où $\text{Cov}(X(s), X(t))$ est la covariance du processus entre les instants s et t (app. A) et H est le paramètre de *Hurst* tel que $\frac{1}{2} < H < 1$ (ce paramètre a été défini à la section 1.4.2).

La modélisation du trafic Internet par un mouvement Brownien fractionnaire (FBM) a été proposée par (Norros, 1995; Norros et Pruthi, 1996). Ainsi, le trafic Internet est modélisé par un processus $\{Y(t); t \geq 0\}$ défini par

$$Y(t) = \mu t + \sqrt{\sigma^2 m} X(t). \quad (1.25)$$

Le processus $Y(t)$ possède trois paramètres μ , σ^2 et H : le paramètre μ représente le débit moyen du trafic, le paramètre σ^2 représente la variance du débit du trafic et $X(t)$ est un processus FBM de paramètre H (le paramètre de *Hurst*).

Le modèle « Bruit fractionnaire Gaussien »

Soit $\{X(t); t \geq 0\}$ un mouvement Brownien fractionnaire. Le processus $\{Z(t)\}$ tel que $Z(t) = X(t + 1) - X(t)$ est appelé *bruit fractionnaire Gaussien* (*Fractional Gaussian Noise* - FGN) et il a le même paramètre de *Hurst* (noté H) que $\{X(t)\}$ (Liu et al., 2006).

Ainsi, le trafic Internet est modélisé par un processus $\{Y(t); t \geq 0\}$ défini par :

$$Y(t) = \mu + \sigma Z(t). \quad (1.26)$$

Le processus $Y(t)$ possède trois paramètres μ , σ^2 et H : le paramètre μ représente le débit moyen du trafic, le paramètre σ^2 représente la variance du débit du trafic et $Z(t)$ est un processus FGN de paramètre H (le paramètre de *Hurst*).

L'avantage des modèles FBM et FGN est la simplicité de leurs formules (éq. 1.25 et éq. 1.26) pour prévoir le débit du trafic, une fois que les différents paramètres sont estimés. Cependant, leur inconvénient est leur incapacité de capturer les dépendances à court terme du trafic (puisque $H > \frac{1}{2}$) (Crovella et Bestavros, 1997).

Le processus ARIMA Fractionnaire (FARIMA)

Le processus ARIMA fractionnaire (FARIMA) est la généralisation naturelle du modèle ARIMA(p, d, q) lorsque le degré de différenciation d peut avoir des valeurs réelles (non seulement des entiers naturels) (Xue et al., 1999; Hosking, 1981a; Krunch et Makowski, 1998).

Une série temporelle $y(t)$ est un processus FARIMA(p, d, q) si

$$(1 - \sum_{i=1}^p \phi_i L^i)(1 - L)^d y(t) = (1 + \sum_{i=1}^q \theta_i L^i) \epsilon(t) \quad (1.27)$$

où ϕ_i et θ_j sont les paramètres du modèle, $\epsilon(t)$ est un bruit blanc (app. A, déf. A.8) de variance σ^2 et L est l'opérateur « retard » (app. A, déf. A.15).

Dans le cas où $p = 0$ et $q = 0$, le processus FARIMA($0, d, 0$) est appelé *fractional differencing noise model* (FDN) (Hosking, 1981a), qui est la forme la plus simple du processus FARIMA. Le paramètre d du processus FARIMA($0, d, 0$) indique le niveau de dépendance à long terme tout comme le paramètre de Hurst H . Il a été démontré que, si $d \in [0, \frac{1}{2}]$, le processus FARIMA($0, d, 0$) présente une dépendance à long terme avec un paramètre de Hurst $H = d + \frac{1}{2}$ (Hosking, 1981b). Le processus FARIMA($0, d, 0$) est

similaire au processus FGN qui peut seulement décrire une dépendance à long terme. Le processus FARIMA(p, d, q) peut être vu comme la combinaison du processus ARMA avec le FARIMA($0, d, 0$). Ainsi, il est capable de modéliser les processus présentant des dépendances à court et à long terme (Crovella et Bestavros, 1997).

(Xue et al., 1999) présentent des directives pour générer du trafic selon le processus FARIMA et pour estimer ses paramètres afin de modéliser et de prévoir un trafic réel. L'inconvénient de ce modèle est la complexité des calculs pour estimer les différents paramètres (Xue et al., 1999).

1.5.4 Les modèles Markoviens

Soit $\{X(k); k \in \mathbb{N}\}$ un processus. L'ensemble des valeurs possibles de $X(k)$ est appelé l'espace d'états (généralement un sous-ensemble de \mathbb{N}). Dans le modèle de trafic Markovien, le changement d'état est interprété comme une nouvelle arrivée (d'un paquet). $\{X(k); k \in \mathbb{N}\}$ est un processus de Markov si sa distribution conditionnelle de probabilité de l'état futur ne dépend que de l'état présent et non pas des états passés. Ainsi, la loi conditionnelle de $X(n+1)$ sachant le passé $\{X(k); 0 \leq k \leq n\}$ ne dépend que de $X(n)$ seulement, c.-à-d. :

$$P[X(n+1) = x | X(0), X(1), X(2), \dots, X(n)] = P[X(n+1) = x | X(n)], \quad (1.28)$$

où x est un état quelconque du processus. Le processus de Poisson (app. A, déf. A.14) est l'un des processus de Markov les plus utilisés.

Ces processus sont très utilisés dans la théorie des systèmes des files d'attente. Un système de file d'attente se décrit par un processus d'arrivée de paquets, un mécanisme de service et une discipline d'attente.

Dans une file d'attente, on suppose que les durées des inter-arrivées sont indépendantes et de même loi. Par exemple, la loi des arrivées peut être à intervalles réguliers, notée D (déterministe) ou elle peut être Poissonienne, notée M (Markov), etc.

Les durées de service sont des variables positives indépendantes et de même loi. La

loi de service peut être de durée constante, notée D (déterministe) ou de durée suivant une loi exponentielle, notée M (Markov), etc.

La discipline d'attente est généralement « premier arrivée, premier servi » (*First In First Out* - FIFO).

Une file d'attente est désignée par le symbole : *Arrivée / Service / Nombre de serveurs / Capacité maximale de la file / Nombre maximal de clients / Discipline de service*. Par défaut, la capacité maximale et le nombre maximal de clients sont égaux à l'infini. La discipline de service est FIFO. Les files d'attente $M/M/1$ et $M/M/1/K$ sont très utilisés pour modéliser la mémoire tampon (*buffer*) dans un routeur. Les formules mathématiques estimant le débit, les temps d'attente et le taux de perte ont été déjà développées et elles sont très utilisées (Nain, 1997).

L'inconvénient des modèles Markoviens est qu'ils sont à mémoire courte : l'état du processus dépend seulement de l'état précédent (eq. 1.28). Ceci est inadéquat pour le trafic caractérisé par une forte corrélation avec le passé et une dépendance à long terme.

Par contre, l'avantage de l'utilisation des processus de Markov par rapport aux autres modèles de trafic est qu'ils permettent, en théorie, de résoudre analytiquement des systèmes de files d'attente.

1.5.5 Le filtre de Kalman

L'objectif du filtre de Kalman (Kalman, 1960; Welch et Bishop, 1995) est d'estimer l'état d'un processus discret $\{Y(t); t \in \mathbb{N}\}$ à valeurs dans \mathbb{R}^n .

On suppose que l'on ne peut pas observer directement les valeurs de $\{Y(t)\}$ (ce processus peut représenter le débit), mais que l'on dispose d'une observation $Z(t) \in \mathbb{R}^m$ (par exemple le débit mesuré). Le but est d'obtenir le plus d'information possible sur

$\{Y(t)\}$. On considère le système dynamique suivant :

$$Y(t) = M_A Y(t-1) + M_B U(t) + W(t-1), \quad (1.29)$$

$$Z(t) = M_H Y(t) + V(t), \quad (1.30)$$

où :

- $\{W(t)\}$ est un bruit blanc Gaussien à valeurs dans \mathbb{R}^n et de matrice de covariance $Q \in \mathbb{R}^{n \times n}$ (app. A, déf. A.10),
- $\{V(t)\}$ est un bruit blanc Gaussien à valeurs dans \mathbb{R}^m et de matrice de covariance $S \in \mathbb{R}^{m \times m}$; $\{V(t)\}$ qui modélise l'erreur d'observation.
- $\{W(t)\}$ et $\{V(t)\}$ et la condition initiale $X(0)$ sont mutuellement indépendants,
- les matrices Q et S sont considérées indépendantes du temps,
- la matrice $M_A \in \mathbb{R}^{n \times n}$ relie l'état précédent $Y(t-1)$ à l'état actuel $Y(t)$ lorsqu'il n'y a pas de bruit. En pratique, M_A est considérée indépendante de t ,
- la matrice $M_H \in \mathbb{R}^{m \times n}$ relie $Y(t)$ à la mesure $Z(t)$. Elle est aussi considérée indépendante de t ,
- $U(t) \in \mathbb{R}^l$ est une entrée optionnelle,
- la matrice $M_B \in \mathbb{R}^{n \times l}$ relie l'entrée optionnelle $U(t)$ à l'état $Y(t)$.

Ainsi, à l'instant t , on recueille une observation $Z(t)$ qui est formée de $M_H Y(t)$ et d'un bruit additif $V(t)$.

On note $\hat{Y}^-(t) \in \mathbb{R}^n$ l'estimation *a priori* à l'instant t sans avoir connaissance de la mesure $Z(t)$, ainsi, $\hat{Y}^-(t)$ représente la prévision de $Y(t)$ à l'instant t . On note $\hat{Y}(t) \in \mathbb{R}^n$ l'estimation *a posteriori* sachant la mesure $Z(t)$.

On définit les erreurs *a priori* et *a posteriori*, notées $e^-(t)$ et $e(t)$ entre $Y(t)$ et les estimations *a priori* et *a posteriori* $\hat{Y}^-(t)$ et $\hat{Y}(t)$ respectivement, par :

$$e^-(t) = Y(t) - \hat{Y}^-(t), \quad (1.31)$$

$$e(t) = Y(t) - \hat{Y}(t). \quad (1.32)$$

Les covariances des erreurs *a priori* et *a posteriori*, notées par $P^-(t)$ et $P(t)$

respectivement, sont données par :

$$P^-(t) = \mathbb{E}[(Y(t) - \hat{Y}^-(t))(Y(t) - \hat{Y}^-(t))^T], \quad (1.33)$$

$$P(t) = \mathbb{E}[(Y(t) - \hat{Y}(t))(Y(t) - \hat{Y}(t))^T], \quad (1.34)$$

où l'exposant T indique l'opérateur de transposition des matrices.

Le filtre de Kalman a deux phases distinctes : *prévision* et *révision*. La phase de prévision utilise la valeur estimée à l'instant précédent, c.-à-d. $Y(t-1)$, pour produire une estimation a priori de $Y^-(t)$ à l'instant t . Cette phase est donnée par ces équations :

$$\hat{Y}^-(t) = M_A \hat{Y}(t-1) + M_B U(t-1), \quad (1.35)$$

$$P^-(t) = M_A P(t-1) M_A^T + Q. \quad (1.36)$$

Dans la phase de révision, les observations de l'instant courant sont utilisées pour corriger l'état prédit dans le but d'obtenir une estimation plus précise. Cette phase est donnée par ces équations :

$$M_K(t) = \dot{P}^-(t) M_H^T (M_H P^-(t) M_H^T + S)^{-1}, \quad (1.37)$$

$$\hat{Y}(t) = \hat{Y}^-(t) + M_K(t) (Z(t) - M_H \hat{Y}^-(t)), \quad (1.38)$$

$$P(t) = (I - M_K(t) M_H) P^-(t), \quad (1.39)$$

où l'exposant -1 indique l'opérateur qui donne l'inverse d'une matrice. La matrice $M_K(t) \in \mathbb{R}^{n \times m}$ est appelée le gain du filtre qui vise à minimiser la covariance a posteriori de l'erreur $P(t)$.

Les paramètres M_A , M_H , M_B , Q et S sont estimés à partir des données disponibles grâce à l'algorithme *espérance-maximisation* (*Expectation-maximization algorithm* - EM) (Dempster, Laird et Rubin, 1977).

Les équations de prévision permettent de calculer $\hat{Y}^-(t)$, la valeur prédite de $Y(t)$, sans avoir connaissance de la valeur de $Z(t)$. Dès que la mesure $Z(t)$ est disponible, les équations de révision sont appliquées. Ainsi, le modèle nécessite la mise à jour (phase de révision) de ses paramètres après chaque phase de prévision. Cet aspect récursif est

l'une des caractéristiques du filtre de Kalman qui permet d'adapter les paramètres du modèle aux changements et à l'évolution du trafic.

L'inconvénient du filtre de Kalman est sa complexité s'il est utilisé pour une prévision à court terme en temps réel puisque la phase de révision est appliquée avant chaque prévision. De plus, le filtre de Kalman suppose que le trafic est Gaussien.

Le filtre de Kalman a été utilisé dans plusieurs travaux pour la prévision du trafic (Jibukumar, Datta et Biswas, 2008; Wang et al., 2006; Zhijun, Yuanhua et Daowen, 2004; Lim et Ab-Hamid, 2000; Soule et al., 2005; Soule et al., 2004).

1.5.6 Le modèle neuroflou α -SNF

Le modèle neuroflou α -SNF est un modèle qui combine la logique floue et les réseaux de neurones (Rouai et Ben.Ahmed, 2001; Zhani, Elbiaze et Kamoun, 2009b). Il se base sur le modèle proposé par (Takagi et Sugeno, 1985). La flexibilité de la logique floue combinée avec la capacité d'apprentissage des réseaux de neurones rendent le modèle plus adaptatif aux caractéristiques du trafic (Zhani, Elbiaze et Kamoun, 2007; Zhani, Elbiaze et Kamoun, 2008; Zhani, Elbiaze et Kamoun, 2009b; Elbiaze et al., 2009; Doulamis, Doulamis et Kollias, 2000; Yu et Chen, 1993).

Le système flou est décrit comme une relation non linéaire entre les entrées x_1, \dots, x_n et la sortie $Y = f(x_1, \dots, x_n)$, où n est le nombre d'entrées x_i . Cette relation est décrite par une collection de règles floues.

Soit c le nombre de règles dans le système flou. Notons par \mathcal{R}_k la $k^{\text{ème}}$ règle où $1 \leq k \leq c$. Le système utilise des règles floues développées par (Takagi et Sugeno, 1985). Une règle floue \mathcal{R}_k est donnée par ce qui suit :

$$\mathcal{R}_k : \text{Si } (x_1, \dots, x_n) \text{ est } A_k \text{ ALORS } y_k \text{ est } b_k, \quad (1.40)$$

où A_k s'appelle un cluster et y_k est la sortie de la règle calculée en utilisant un réel b_k . Dans la logique floue, chaque point x appartient à un cluster A avec un degré d'appartenance d'une valeur comprise entre zéro et un. Cette valeur est obtenue grâce à une

fonction d'appartenance notée $\mu_A(x)$. Ainsi, chaque règle \mathcal{R}_k évalue le degré d'appartenance de chaque élément (x_1, \dots, x_n) au cluster A_k grâce à une fonction d'appartenance notée $\mu_{A_k}(x_1, \dots, x_n)$. Ensuite, la sortie y_k est évaluée en fonction de ce degré d'appartenance par :

$$y_k = \mu_{A_k}(x_1, \dots, x_n) \cdot b_k. \quad (1.41)$$

La règle \mathcal{R}_k peut être écrite sous une autre forme :

$$\mathcal{R}_k : \text{Si } x_1 \text{ est } A_{1k} \text{ et } x_i \text{ est } A_{ik}, \dots, \text{ et } x_n \text{ est } A_{nk} \text{ ALORS } y_k \text{ est } b_k \quad (1.42)$$

où le cluster A_{ik} est la projection de A_k sur la $i^{\text{ème}}$ dimension. On note par $\mu_{A_{ik}}(x_i)$ la fonction d'appartenance de x_i au cluster A_{ik} . Ainsi, la fonction $\mu_{A_k}(x_1, \dots, x_n)$ est donnée par :

$$\mu_{A_k}(x_1, \dots, x_n) = \prod_{i=1}^n \mu_{A_{ik}}(x_i). \quad (1.43)$$

La fonction d'appartenance $\mu_{A_{ik}}(x_i)$ est donnée par (Lin et Cunningham, 1995) :

$$\mu_{A_{ik}}(x_i) = \exp(-|w_{ikg}x_i + w_{ikc}|^{l_{ik}}), \quad (1.44)$$

où w_{ikg} , w_{ikc} et l_{ik} sont des paramètres utilisés pour ajuster la forme générale de la fonction. Cette fonction a été choisie étant donnée sa dérivabilité, sa forme paramétrée et générique qui permet d'approcher des fonctions d'appartenance triangulaires, trapézoïdales etc.

Finalement, la sortie du système Y est donnée par :

$$Y = \frac{\sum_{k=1}^c y_k}{\sum_{k=1}^c \mu_{A_k}(x_1, \dots, x_n)}. \quad (1.45)$$

Le modèle flou est alors incorporé dans un réseau de neurones équivalent. La figure 1.2 montre un exemple du modèle α -SNF utilisant trois règles et deux entrées. Celui-ci a deux entrées x_1 et x_2 . Les paramètres w_{ikg} et w_{ikc} sont les poids des connexions. Chaque nœud A_{ik} calcule la fonction d'appartenance $\mu_{A_{ik}}(x_i)$ et chaque nœud A_k calcule $\mu_{A_k}(x_1, x_2)$. Le nœud de sortie calcule la sortie du système Y .

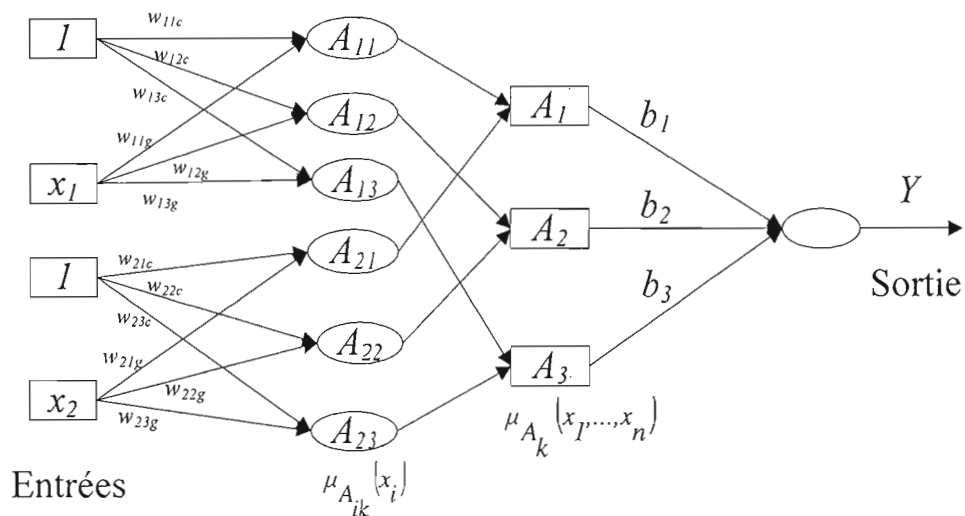


Figure 1.2 Exemple d'un réseau de neurones (trois règles et deux entrées).

Les paramètres w_{ikg} , w_{ikc} , l_{ik} et b_k sont initialisés en utilisant une méthode appelée « *semi- α -coupe densité* » proposée par (Rouai et Ben.Ahmed, 2001). Ils sont ensuite optimisés grâce à un *algorithme d'apprentissage*. L'apprentissage consiste à fournir au système un ensemble de données en entrée et leurs sorties désirées. Le système calcule la sortie correspondante à chaque entrée. Ensuite, ses paramètres sont modifiés afin de réduire l'erreur entre la sortie calculée et la sortie désirée. Ce procédé est itéré plusieurs fois afin d'optimiser les paramètres.

L'algorithme d'apprentissage, que nous avons utilisé, est la *rétropropagation du gradient* (Rouai et Ben.Ahmed, 2001; Lippmann, 1987; Rumelhart, Hinton et Williams, 1986).

Le nombre de règles utilisées et le nombre d'itérations pour l'algorithme d'apprentissage sont des paramètres à spécifier. Une étude sur ces paramètres a été effectuée dans (Zhani, 2005) en se basant sur des traces réelles de trafic Internet. Ainsi, nous utilisons les valeurs qui ont donné les meilleurs résultats, à savoir trois règles floues et 100 itérations pour la phase d'apprentissage.

1.5.7 Conclusion sur les modèles de prévision

Chaque modèle de prévision présente des avantages et des inconvénients. L'apparition du caractère LRD du trafic a conduit plusieurs chercheurs à abandonner les modélisations et les hypothèses Markoviennes en faveur de nouveaux modèles plus complexes pour le trafic. Par conséquent, les recherches récentes se sont orientées vers les modèles tenant compte de l'auto-similarité du trafic (tels que FARIMA et FBM). Toutefois, peu de travaux ont exploité ces modèles auto-similaires pour des applications concrètes de conception de protocoles ou de planification des réseaux. En effet, il est difficile de manipuler ces modèles vu la complexité de leurs structures mathématiques ainsi que leur complexité en termes de calcul. De plus, ces modèles n'améliorent pas la prévision en termes d'erreur par rapport aux modèles linéaires qui sont plus simples tels que le modèle ARIMA (Qiao, Skicewicz et Dinda, 2003; Qiao, Skicewicz et Dinda, 2002).

D'autre part, pour la majorité des modèles de prévision, il est difficile de comprendre ou d'établir une relation entre les coefficients du modèle et le trafic réel (la famille ARIMA, FARIMA, le filtre de Kalman, modèle neuroflou).

Toutes ces approches visent à obtenir une prévision assez précise en utilisant différentes techniques pour capturer les caractéristiques du trafic réel. Par conséquent, dans le présent travail, nous avons choisi de considérer le débit des données comme étant une série temporelle. Nous utilisons les modèles ARMA, ARIMA et LMMSE vu leur simplicité en termes de complexité du calcul et précision en termes d'erreur de la prévision. Nous exploitons aussi le modèle neuroflou α -SNF qui est capable de capturer avec précision les caractéristiques du trafic grâce à sa capacité d'apprentissage.

Dans ce qui suit, nous présentons les travaux effectués dans le domaine de la prévision du trafic.

1.6 Revue de la littérature

Durant la dernière décennie, beaucoup de laboratoires ont lancé des projets de recherche portant sur *la métrologie de l'Internet*, ou science de mesure de l'Internet. Ces projets visent à mesurer le trafic Internet, à l'analyser et à le modéliser (Projet NIMI⁴ (Paxson, Adams et Mathis, 2002), Projet RIPE⁵, le projet américain AMP⁶ de NLANR⁷, le projet IPMON de Sprint⁸). Les mesures couvrent les différents paramètres caractérisant le réseau tels que le débit, le taux de perte, les délais, etc. Ainsi, l'objectif final est d'avoir une compréhension plus profonde des différents paramètres qui caractérisent le réseau Internet et de suivre de plus près leur évolution et sa répercussion sur la qualité de service offerte aux clients. Ces mesures permettent aussi la modélisation du trafic, l'étude du comportement des différents protocoles (IP, TCP, UDP etc.), l'évaluation des performances du réseau et le développement d'outils d'analyse du réseau (Barakat et al., 2003; Papagiannaki et al., 2003b; Owezarski et Larrieu, 2004; Scherrer et al., 2006; Papagiannaki et al., 2003a; John, Tafvelin et Olovsson, 2010; Chabchoub et al., 2010). Elles permettent de mettre au point et optimiser les protocoles existants et de prévoir le trafic et les capacités requises dans le futur. De plus, l'analyse des mesures permet la détection des anomalies et des attaques dans le réseau (Papagiannaki et al., 2005; Cantieni et al., 2006; Markopoulou et al., 2008; Farraposo, Owezarski et Monteiro, 2006; Larrieu et Owezarski, 2005; Janowski et Owezarski, 2010).

Nous commençons par présenter les travaux qui ont étudié la performance des différents modèles de prévision. Ensuite, nous présentons les mécanismes de contrôle de congestion des files d'attente, notamment ceux qui ont exploité la prévision du trafic

4. NIMI : National Internet Measurement Infrastructure

5. RIPE : RIPE Network Coordination Centre - <http://www.ripe.net>

6. AMP : Active Measurement Project - <http://www.nlanr.net/PMA/>

7. NLANR : National Laboratory for Applied Network Research - <http://www.nlanr.net/>

8. The Academic Research group at Sprint - <https://research.sprintlabs.com/>

pour effectuer ce contrôle.

1.6.1 Prédiction du trafic

Le premier système de mesure et de prédiction pour le trafic Internet a été proposé par (Wolski, 1997). Il consiste à appliquer simultanément trois modèles de prédiction (la moyenne; la médiane et le modèle autorégressif (AR)) pour sélectionner automatiquement et dynamiquement le meilleur d'entre eux afin de l'utiliser. L'idée de modifier dynamiquement le modèle de prédiction est intéressante, puisque les caractéristiques du trafic diffèrent en termes de temps et d'espace (c.-à-d. d'un moment à un autre ou d'un réseau à un autre).

Une autre étude a été réalisée par (Sang et al Li, 2000; Sang et al Li, 2002) pour analyser la performance de la prédiction du trafic dans le réseau Internet. L'approche considère que la prédiction du débit d'un lien du réseau est basée sur deux paramètres, à savoir l'horizon de la prédiction (l'intervalle pour lequel la prédiction est effectuée) et l'évolution de l'erreur en fonction de cet horizon. Les auteurs ont utilisé deux modèles stationnaires : le modèle autorégressif à moyenne mobile (ARMA) et le processus *Markov modulated Poisson process* (MMPP). Leur étude a montré que la précision de la prédiction est détériorée lorsque l'horizon augmente. À partir des études analytiques et empiriques, ils ont étudié le rôle de l'application des méthodes statistiques sur le trafic lors de la prédiction. En particulier, ils ont trouvé que l'échantillonnage et le lissage ont des effets positifs sur la prédiction. Ils ont aussi remarqué que les meilleurs résultats de la prédiction sont obtenus si les petites variations du trafic sont filtrées. Ces variations peuvent être filtrées étant donné qu'elles sont moins importantes pour l'allocation de la bande passante.

D'autres travaux se sont intéressés à la prédiction à court terme et ont étudié l'effet de la granularité (l'échelle du temps), le degré de lissage sur la précision de la prédiction du trafic du réseau (Qiao, Skicewicz et Dinda, 2002; Qiao, Skicewicz et Dinda, 2003). Ils ont appliqué plusieurs modèles de série temporelle (MEAN, LAST,

AR, MA, ARIMA, FARIMA) sur un ensemble de traces de paquets. Les prévisions ont été effectuées en considérant différentes échelles de temps. La performance a été mesurée par le ratio *Average Relative Variance* (ARV) obtenu par la division de l'*erreur quadratique moyenne* de la prévision (*Mean Squared Error* - MSE) par la variance des données. Les auteurs ont remarqué que la précision de la prévision dépend du cas étudié, c'est-à-dire, qu'elle varie d'une trace à une autre. L'étude a montré que, contrairement aux résultats trouvés par (Sang et Qi Li, 2000; Sang et Qi Li, 2002), la précision de la prévision n'augmente pas lorsqu'on augmente le degré de lissage des données. En effet, dans la majorité des traces étudiées, il y a un degré de lissage, appelé « *sweet spot* », pour lequel l'erreur est minimisée et la précision de la prévision est clairement meilleure. En dépassant ce point, l'erreur de la prévision augmente. Cependant, l'étude n'a pas réussi à déterminer la raison pour laquelle la prévision est améliorée pour ce point en particulier. D'autre part, pour une prévision à court terme (des millisecondes jusqu'aux minutes), la même étude a montré que les modèles FARIMA et ARIMA donnent la même performance en termes d'erreur que le modèle ARMA classique (Qiao, Skicewicz et Dinda, 2002; Qiao, Skicewicz et Dinda, 2003). Ceci peut être expliqué par le fait que le trafic peut être considéré stationnaire au sens large à très faible échelle de temps (des millisecondes jusqu'aux minutes) (Zhang et Duffield, 2001).

(Bermolen et Rossi, 2009) ont testé l'utilisation du modèle *Support Vector Regression* (SVR) pour prévoir la charge d'un lien. Ils ont été motivés par le succès de ce modèle qui vient du domaine de la théorie de l'apprentissage statistique. Ainsi, ils l'ont comparé empiriquement au modèle autorégressif (AR) et au modèle de la moyenne mobile (MA) en termes de précision de la prévision et en termes de complexité. La comparaison a couvert la phase d'apprentissage (définition des paramètres du modèle) et la phase de prévision. Leurs résultats montrent que bien que le modèle SVR améliore légèrement la précision de la prévision, il est plus gourmand en termes de complexité.

Par conséquent, pour des prévisions à court terme, les résultats de ces travaux encouragent à ne pas considérer des modèles complexes. En effet, ils n'offrent pas une meilleure performance lorsqu'ils sont comparés aux modèles classiques tels que

les modèles autorégressifs à moyenne mobile (ARMA) (Qiao, Skicewicz et Dinda, 2002; Qiao, Skicewicz et Dinda, 2003; Bermolen et Rossi, 2009; Garroppo et al., 2008).

D'autres travaux ont étudié la prédictibilité du débit d'une connexion TCP (He, Dovrolis et Ammar, 2005; He, Dovrolis et Ammar, 2007). Les techniques de prévision du trafic au niveau de la couche TCP (c.-à-d. la prévision de la quantité de données reçue par la couche TCP) ont été classées en deux catégories : une catégorie de modèles basés sur des formules mathématiques (*Formula-Based* - FB) et une catégorie de modèles basés sur l'historique du trafic (*History-Based* - HB). Les modèles FB sont des modèles mathématiques qui expriment le trafic TCP en fonction des caractéristiques du chemin du réseau étudié (le temps d'aller-retour, nombre de flux etc.). Les techniques HB prévoient le trafic d'un chemin dans le réseau en se basant sur l'historique du trafic lorsque celui-ci est disponible. Ils ont montré que les techniques HB sont précises, mais elles sont dépendantes du chemin ou du lien analysé. Les méthodes FB sont précises seulement si le chemin ou le lien étudié n'est pas saturé. Il est important aussi de noter que les modèles HB peuvent être appliqués indépendamment du protocole utilisé au niveau transport. Ils peuvent aussi être appliqués au niveau de n'importe quelle couche OSI (*Open System Interconnection* - OSI), notamment la couche IP. Pour ces raisons, nous utilisons les modèles HB dans le présent travail.

À partir de ces travaux, nous pouvons aussi classer les modèles de prévision en deux catégories : les modèles ayant une phase d'apprentissage (*Training-Based* - TB) et les modèles sans phase d'apprentissage (*Non-Training-Based* - NTB).

La phase d'apprentissage est une phase pendant laquelle les paramètres du modèle sont ajustés en fonction de l'historique des données disponibles. L'algorithme d'apprentissage est généralement exécuté une seule fois pour définir les paramètres du modèle. Le modèle est ensuite capable de générer la prévision en fonction de quelques retards (anciennes observations).

Les techniques NTB n'ont pas besoin d'historique ou d'algorithme d'apprentissage. Généralement, ces modèles calculent la valeur prédite directement et seulement

en fonction de quelques retards (anciennes observations). Les modèles neuroflous et les modèles ARIMA sont des modèles TB. Le modèle LMMSE est un exemple de modèle NTB.

D'autres travaux ont exploité la prévision pour la planification des capacités futures du réseau. Une première tentative pour une prévision à long terme d'un réseau IP a été effectuée par (Groschwitz et Polyzos, 1994). Les auteurs ont calculé le nombre total de paquets observé pour tous les nœuds du réseau fédérateur *NSFNET*. Ils ont montré que la série temporelle obtenue peut être modélisée avec le modèle ARIMA. Ils ont essayé de prévoir la charge totale du réseau fédérateur pour une année puis pour deux années dans le futur en utilisant l'historique des données des quatre années antérieures. Le modèle a donné des résultats assez précis pour une année dans le futur. Cependant, prévoir une seule valeur pour la charge d'un réseau est insuffisant pour planifier le changement des capacités de chaque lien du réseau. En effet, on a besoin de prévoir dans quel lien du réseau des surcharges pourront survenir ; ceci permet d'identifier les liens où des approvisionnements en ressources seront requis.

Une approche de prévision à long terme a été présentée par (Papagiannaki et al., 2003b; Papagiannaki et al., 2005). Elle permet de prévoir à quel moment un lien doit être ajouté ou amélioré dans un réseau fédérateur IP. Ils ont utilisé les statistiques SNMP⁹ collectées continuellement depuis 1999 jusqu'à 2002. Ils ont agrégé la charge entre chaque paire de PoP¹⁰ adjacents et ils ont étudié leurs évolutions dans une échelle de temps supérieure à une heure. Ils ont démontré que le trafic des réseaux fédérateurs IP présente des tendances à long terme, une forte périodicité et une variabilité à différentes échelles de temps. La méthodologie de prévision des auteurs s'appuie sur l'analyse multirésolution par ondelettes et le modèle de série temporelle ARIMA. En effet, les auteurs appliquent un lissage sur les données jusqu'à l'obtention de la tendance à long terme

9. SNMP (*Simple Network Management Protocol*) : protocole dédié aux tâches de surveillance et d'administration d'un réseau.

10. PoP (Point of Presence) est une interface réseau entre deux entités de communication

du trafic. Les fluctuations autour de la tendance obtenue sont analysées à différentes échelles de temps. Ils ont montré que la plus importante variabilité est due à la fluctuation à l'échelle de 12 heures. Les auteurs ont démontré que la charge hebdomadaire entre deux nœuds du réseau peut être modélisée par le modèle ARIMA avec une bonne précision : la prévision de la tendance à long terme et les fluctuations du trafic à l'échelle de 12 heures donne une estimation précise pour au moins six mois dans le futur.

(He et al., 2002) ont exploité la prévision du trafic afin d'effectuer un contrôle plus performant de la congestion au niveau du protocole TCP. En effet, ils ont montré que la dépendance à long terme pourrait être détectée en temps réel et utilisée pour la prévision du trafic pour un horizon de prévision d'au moins un temps d'aller-retour (RTT). Le modèle de prévision utilisé est le modèle LMMSE ayant les 20 dernières mesures (retards) comme des variables en entrée. Ils ont défini un nouveau protocole appelé *Transmission Control Protocol with Traffic Prediction* (TCP/TP), où le mécanisme de contrôle de congestion de TCP ajuste sa fenêtre d'anticipation (en augmentant ou en diminuant sa taille) en considérant la bande passante disponible dans le futur. Par conséquent, la connexion TCP n'utilise pas aveuglement toute la bande passante disponible, mais elle essaie de prévoir la congestion. Les auteurs ont montré que l'impact de l'erreur de la prévision est négligeable sur l'efficacité du protocole TCP/TP.

(Garroppo et al., 2008) ont proposé une approche pour l'allocation des ressources basée sur la prévision du trafic. Deux modèles de prévision ont été considérés. Le premier est un modèle non-linéaire appelé *radial basis function predictor* (RBFP). Le deuxième est le modèle *Normalized least mean square Predictor* (NLMS) qui est un modèle linéaire simple. Les deux modèles sont comparés empiriquement selon trois critères, à savoir, la complexité, la précision et la rapidité de l'adaptation du modèle aux variations du trafic. Les résultats montrent que, bien que le NLMS soit un modèle de complexité moyenne comparé au RBFP, il est capable d'atteindre les mêmes performances en termes de précision et d'adaptation aux variations.

Plusieurs chercheurs ont travaillé sur la prévision du trafic vidéo à débit variable

(*Variable Bit Rate* - VBR) (Jibukumar, Datta et Biswas, 2008; Wang et al., 2006; Zhi-jun, Yuanhua et Daowen, 2004; Doulamis, Doulamis et Kollias, 2000; Krunz et Makowski, 1998). Ces travaux utilisent les modèles de prévision des séries temporelles tels que les modèles autorégressifs à moyenne mobile, les filtres de Kalman ou le modèle LMMSE. La prévision du trafic vidéo est souvent exploitée pour l'allocation dynamique des ressources pour les flux multimédias.

L'exploitation de la prévision à court terme pour améliorer les mécanismes de contrôle de congestion (*Active Queue Management* - AQM) a été le sujet d'autres travaux. Ainsi, nous présentons, dans ce qui suit, les principaux mécanismes de contrôle de congestion, notamment ceux qui se basent sur la prévision du trafic.

1.6.2 Les mécanismes de gestion des files d'attente

Les mécanismes de gestion des files d'attente (*Active Queue Management* - AQM) visent principalement à éviter la congestion des files d'attente. Ils se basent sur le rejet proactif (à l'avance) des paquets pour prévenir les sources de la congestion du réseau.

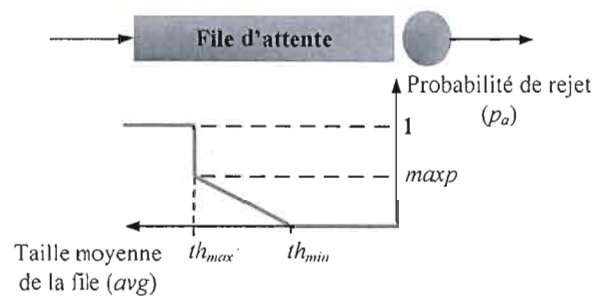


Figure 1.3 Mécanisme RED.

Le mécanisme *Random Early Detection* (RED) est l'une des solutions les plus connues pour résoudre le problème de congestion des files d'attente (Floyd et Jacobson, 1993). Chaque fois qu'un paquet arrive, le mécanisme RED met à jour la taille moyenne de la file d'attente (notée avg) (fig. 1.3). Si la taille moyenne dépasse un seuil minimal (th_{min}), le mécanisme rejette le paquet avec une probabilité de rejet p_a . Cette probabilité

est calculée en fonction de la taille moyenne de la file d'attente *avg*. Elle atteint la probabilité maximale de rejet *maxp* lorsque la taille moyenne atteint un seuil maximal (th_{max}). Si la taille maximale dépasse le seuil maximal, tous les paquets sont rejetés automatiquement. L'algorithme RED est décrit dans l'appendice B.

Le mécanisme RED a été conçu pour accompagner un protocole de contrôle de congestion au niveau de la couche transport tel que TCP. Ses avantages se résument principalement à prévenir la congestion des files d'attente et à éviter la synchronisation entre les différents flux TCP (Floyd et Jacobson, 1993). Il a été recommandé par l'IETF¹¹ (Braden et al., 1998) pour être déployé dans les réseaux. Cependant, des recherches plus récentes ont découvert plusieurs inconvénients de RED. L'algorithme RED n'est pas capable de stabiliser la taille de la file d'attente autour d'une valeur cible (Xu, Wang et Wang, 2005). En effet, la taille moyenne de la file dépend de la charge du trafic et des paramètres de configuration de RED (Sun et al., 2003). L'algorithme RED original peut mener à une sous-utilisation du lien (Hollot et al., 2001). De plus, RED ne garantit pas une équité entre les flux (Lin et Morris, 1997).

Afin de pallier les inconvénients de RED, plusieurs mécanismes AQM ont été proposés, par exemple, PI (Hollot et al., 2001), REM (Athuraliya et al., 2001), State Feedback (Gao et Hou, 2003), AVQ (Gao et Hou, 2001), BLUE (Wu, Shi et Kandlur, 2002), SRED (Ott, Lakshman et Wong, 1999), PURPLE (Pletka, Waldvogel et Mannel, 2003), PAQM (Gao, He et Hou, 2002), FRED (Lin et Morris, 1997), BRED (Anjum et Tassiulas, 1999), APACE (Jain, Karandikar et Verma, 2003; Jain, Karandikar et Verma, 2004), AFRED (Wang et al., 2003) et les mécanismes basés sur la logique floue (FAFC) (Aoul, Mehaoua et Skianis, 2007).

Les mécanismes peuvent être classés en trois groupes selon leurs objectifs (Gao, He et Hou, 2002) :

11. Détachement d'ingénierie d'Internet (*The Internet Engineering Task Force* - IETF) est un groupe informel, international, ouvert à tout individu, qui participe à l'élaboration de standards pour Internet.

- les mécanismes qui visent à atteindre une équité entre les différents flux tels que FRED, BRED.
- les mécanismes qui permettent d’atteindre simultanément une importante utilisation du lien et un taux de perte de paquets assez faible tels que PI, REM, AVQ, BLUE, PURPLE.
- les mécanismes qui permettent de stabiliser la taille de la file d’attente tels que SRED, PAQM, ASPACE, AFRED

Chacun des mécanismes AQM mentionnés a sa propre méthode pour détecter la congestion et calculer la probabilité de rejet appropriée.

Le mécanisme PURPLE est un AQM basé sur la prévision du trafic (Pletka, Waldvogel et Mannel, 2003). Ce mécanisme suppose que les paquets ne sont pas rejetés, mais, plutôt marqués. Ainsi, chaque routeur peut avoir des statistiques sur l’état de congestion du réseau en comptant le nombre des paquets marqués qui le traversent. Cette information est utilisée pour prévoir la réaction des sources TCP afin de contrôler les files d’attente. Le mécanisme PURPLE offre un bon équilibre entre le débit effectif (c.-à-d. le débit perçu par la couche application au niveau de la destination), le débit du lien et le délai moyen de bout en bout. Cependant, il emploie quelques paramètres difficiles à évaluer dans un réseau réel, tels que la probabilité qu’un paquet soit marqué par un nœud précédent ou l’estimation de la valeur du RTT.

Le mécanisme PAQM (*Predictive AQM*) est un autre AQM qui exploite la possibilité de prévoir le trafic et son caractère de dépendance à long terme pour augmenter la performance du AQM (Gao, He et Hou, 2002). Au début de chaque intervalle de taille I , le modèle de prévision LMMSE est appliqué pour prédire dynamiquement la taille de la file d’attente à la fin de l’intervalle. La valeur prédite est utilisée pour calculer la probabilité de rejet à considérer pour l’intervalle en question. La probabilité de rejet

des paquets dans la file à l'intervalle $k + 1$, notée $p(k + 1)$, est donnée par :

$$p(k + 1) = \begin{cases} 0 & Q(k) < C + \bar{Q} - \hat{f}(k + 1) \\ \frac{Q(k) + \hat{f}(k + 1) + C - \bar{Q}}{\hat{f}(k + 1)} & C + \bar{Q} - \hat{f}(k + 1) < Q(k) < C + \bar{Q} \\ 1 & Q(k) > C + \bar{Q} \end{cases} \quad (1.46)$$

où $\hat{f}(k + 1)$ est la valeur prédite de la taille de la file (exprimée en nombre de paquets) à la fin de l'intervalle $(k + 1)$, $Q(k)$ est la taille de la file d'attente à la fin de l'intervalle k , \bar{Q} est la valeur désirée comme taille de la file d'attente et C est le nombre maximal de paquets qui peuvent être servis pendant un intervalle de taille I .

Les résultats des simulations du mécanisme PAQM ont prouvé qu'il stabilise la taille de la file d'attente et qu'il améliore l'utilisation du lien sans causer de perte excessive de paquets.

Un autre mécanisme AQM basé sur la prévision a été baptisé *Adaptive prediction based approach for congestion estimation* (APACE) (Jain, Karandikar et Verma, 2003; Jain, Karandikar et Verma, 2004). Il utilise la prévision de la longueur instantanée de la file d'attente après la réception de $N0$ paquets, notée $\hat{q}(n + N0)$. Le modèle de prévision utilisé est le *Normalized Least Mean Square* (NLMS) qui effectue la prévision en utilisant les n derniers retards. À la réception de chaque paquet, la probabilité de rejet p est calculée par :

$$p = \begin{cases} 0, & \text{si } \hat{q}(n + N0) \leq \alpha B \\ \frac{\max(p(\hat{q}(n + N0) - \alpha B), 0)}{(1 - \alpha)B}, & \text{si } \alpha B < \hat{q}(n + N0) \leq B \\ 1, & \text{si } \hat{q}(n + N0) > B \end{cases} \quad (1.47)$$

où B est la taille maximale de la mémoire tampon (*buffer*), α est un paramètre à fixer tel que $0 < \alpha < 1$ et \max est la probabilité maximale de rejet fixée par l'administrateur. Les simulations réalisées montrent que le mécanisme APACE donne une meilleure performance par rapport à PAQM en termes d'utilisation du lien et qu'il s'adapte plus rapidement aux changements du trafic.

Tableau 1.1 Résumé des différents AQM

Objectif	AQM	Indice de Congestion	Basé sur la prévision	Utilisation de la prévision
L'équité entre les flux	FRED	Taille de la file	Non	-
	BRED	Taille de la file	Non	-
Une utilisation importante du lien et un faible taux de perte	BLUE	Taille de la file	Non	-
	REM	Taille de la file,	Non	-
		débit d'entrée	Non	-
	PI	Taille de la file,	Non	-
		débit d'entrée	Non	-
	AVQ	Débit d'entrée	Non	-
La stabilisation de la taille instantanée de la file	PURPLE	Taille de la file	Oui	Probabilité de rejet
	α -SNFAQM	Congestion pour deux intervalles consécutifs	Oui	Classification de la congestion
	SRED	Taille de la file	Non	-
	PAQM	Taille de la file	Oui	Probabilité de rejet
	APACE	Taille de la file	Oui	Probabilité de rejet
	AFRED	Taux de perte des paquets	Oui	Probabilité de rejet
	α -SNFAQM	Prévision du débit d'entrée	Oui	Classification de la congestion

Par ailleurs, un autre moyen de classer les mécanismes AQM existants est de les séparer en fonction de l'utilisation de la prévision du trafic pour prévoir la congestion. Le tableau 1.1 montre les différents AQM classés selon leurs objectifs. Il montre l'indice permettant de détecter ou prévoir la congestion pour chacun d'eux. Pour le cas où la prévision du trafic est utilisée, le tableau précise comment la prévision a été utilisée pour le mécanisme en question. Il compare également les mécanismes existants au mécanisme

α - SNFAQM proposé dans le chapitre 4.

Dans notre travail la prévision est appliquée aussi pour le contrôle du taux de perte dans les réseaux optiques à commutation de rafales. Ainsi, nous présentons ces réseaux et les différents problèmes reliés à la contention et la perte des paquets.

1.7 Les réseaux optiques à commutation de rafales

Les réseaux optiques à commutation de rafales (*Optical Burst Switching* - OBS) constituent une nouvelle génération de réseaux optiques (Jue et Vokkarane, 2005). Les réseaux OBS se situent entre les réseaux à commutation de circuits et les réseaux à commutation de paquets afin de tirer profit des avantages de chacun de ces types de réseaux. En effet, les réseaux OBS permettent d'éviter les réservation inutiles des ressources par rapport aux réseaux à commutation de circuit. Ils permettent aussi de réduire les temps de traitement électronique dans les nœuds comparés aux réseaux à commutation de paquets.

1.7.1 Architecture du réseau OBS

Un réseau OBS est un réseau optique qui commute des rafales (*bursts*) au lieu des paquets. Une rafale (*burst*) est un ensemble de paquets qui sont assemblés dans une seule entité. Le réseau réserve les ressources seulement avant l'envoi de la rafale et les libère juste après la transmission.

Un réseau OBS est constitué de nœuds (routeurs ou stations) reliés par des liens de fibres optiques (fig. 1.4). À l'intérieur du réseau, la transmission est totalement effectuée avec de la fibre optique. Chaque lien de fibre optique est capable de faire passer plusieurs signaux sur des canaux de longueurs d'onde différentes utilisant le multiplexage en longueur d'onde (*Wavelength Division Multiplexing* - WDM).

On distingue deux types de nœuds :

- Les nœuds de cœur (*core nodes*) sont des nœuds qui se trouvent à l'intérieur

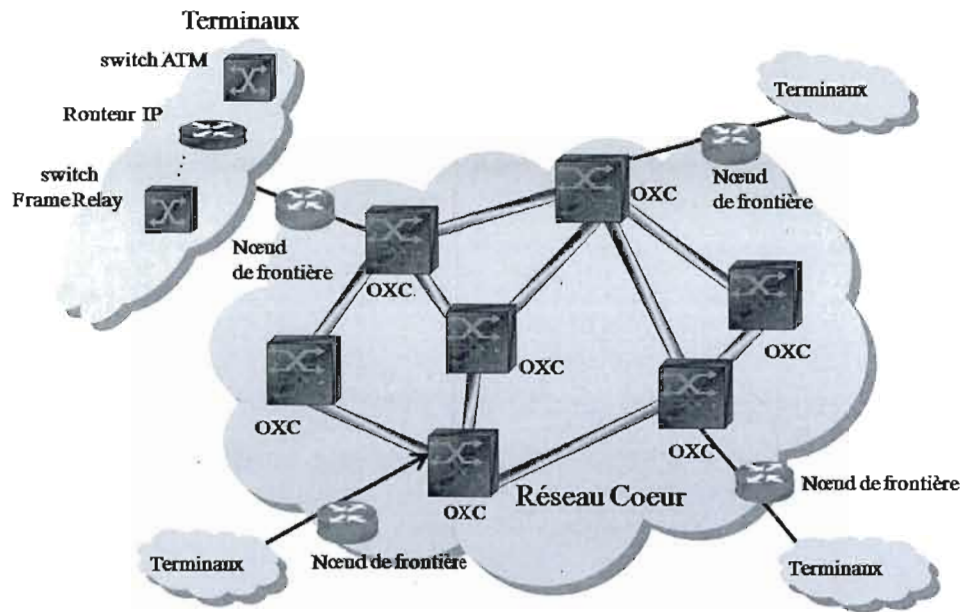


Figure 1.4 Architecture d'un réseau OBS.

d'un réseau OBS. Un nœud de cœur a pour rôle de commuter les rafales à l'intérieur du réseau et de gérer la contention entre les rafales.

- Les nœuds de frontière (edge nodes) sont des nœuds qui se trouvent à la frontière du réseau. Le rôle d'un nœud de frontière est de réaliser l'assemblage des rafales (*la burstification*) qui consiste à assembler un certain nombre de paquets dans une seule entité appelée *rafale* (*burst*). À la réception d'une rafale, le nœud de frontière effectue l'opération inverse qui consiste à désassembler la rafale pour retrouver les paquets qui la constituent. Ainsi, on peut distinguer deux types de nœuds de frontière : les nœuds d'entrée (*ingress node*) qui assemblent les paquets pour construire les rafales et les nœuds de sortie (*egress node*) qui désassemblent les rafales. Les nœuds de frontières sont en même temps des nœuds d'entrée et de sortie selon les directions des flux de paquets. Le nœud d'entrée qui crée la rafale est aussi responsable de choisir la longueur d'onde à utiliser pour la rafale et de planifier son envoi vers la destination.

1.7.2 Assemblage des rafales

Il existe plusieurs algorithmes pour l'assemblage des rafales (*la burstification*). La majorité des algorithmes d'assemblage est basée soit sur un temporisateur (*timer-based*) (Ge, Callegati et Tamil, 2000), soit sur la taille de la rafale (*threshold-based*) (Vokkarane, Haridoss et Jue, 2002; Vokkarane et al., 2002). Pour les approches utilisant le temporisateur, la rafale est créée et envoyée dans le réseau périodiquement (la période est fixée par l'administrateur). Ainsi, les rafales ont des tailles différentes. Pour les approches basées sur la taille, une limite est définie pour la taille de la rafale (en KB ou en nombre de paquets contenus dans la rafale).

1.7.3 Signalisation et réservation des ressources

Afin d'envoyer une rafale à travers les nœuds de cœur, un mécanisme de signalisation doit être implémenté pour allouer les ressources nécessaires et configurer les commutateurs optiques au niveau de chaque nœud.

Avant d'envoyer la rafale, un *paquet de contrôle* est envoyé afin d'effectuer la réservation d'une longueur d'onde (c.-à-d. *un canāl*). Les canaux sont divisés en deux groupes : les canaux de contrôles qui sont dédiés à l'envoi des paquets de contrôle et les canaux de données qui sont dédiés à l'envoi des rafales (C. Qiao, 2000; Qiao et Yoo, 1999b; Vokkarane, Jue et Sitaraman, 2002).

L'intervalle de temps qui sépare l'envoi du paquet de contrôle et l'envoi de la rafale correspondante est appelé *l'offset* (Detti, Eramo et Listanti, 2002; Qiao et Yoo, 1999a).

Plusieurs mécanismes de réservation de longueur d'onde ont été proposés pour les réseaux OBS, notamment *just-in-time* (JIT) (Baldine et al., 2002), *Horizon* (Turner, 1999) et *just-enough-time* (JET) (Qiao et Yoo, 1999b). On peut classer ces mécanismes en deux catégories (Teng et Rouskas, 2003) :

- Les mécanismes avec réservation immédiate : si le canal est libre, la longueur

d'onde est immédiatement réservée pour la rafale dès l'arrivée du paquet de contrôle correspondant (fig. 1.5). Ainsi, aucune autre réservation ne peut être effectuée jusqu'à l'envoi de la rafale. En d'autres termes, la réservation immédiate ne permet qu'une seule réservation à la fois. Si le canal n'est pas libre à l'arrivée du paquet de contrôle (c.-à-d. il est déjà réservé), celui-ci est rejeté et la rafale correspondante est rejetée aussi à son arrivée. Le mécanisme JIT utilise la réservation immédiate.

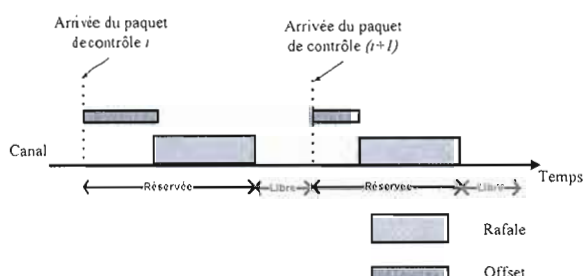


Figure 1.5 Réserve immédiate (JIT).

- Les mécanismes avec réservation différée : la longueur d'onde n'est pas réservée immédiatement à l'arrivée du paquet de contrôle. Elle est réservée juste avant l'arrivée du premier bit de la rafale. Contrairement à la réservation immédiate, si la longueur d'onde n'est pas libre à l'instant où le paquet de contrôle est reçu, mais libre lorsque la rafale arrivera, la longueur d'onde sera quand même réservée pour la nouvelle rafale. Ainsi, la réservation différée permet à plusieurs paquets de contrôle d'effectuer plusieurs réservations en même temps (à condition qu'il n'y ait pas de chevauchement entre les rafales correspondantes). Les mécanismes Horizon et JET utilisent la réservation différée.

La figure 1.6 montre un canal où la rafale i a déjà été planifiée (c'est-à-dire la réservation a déjà été effectuée). La rafale $(i + 2)$ peut être planifiée grâce à la réservation différée. Ceci n'est pas possible avec un mécanisme de réservation immédiate.

On remarque que le canal est libre pendant *un intervalle vacant* c.-à-d. l'intervalle de temps entre l'instant actuel et l'instant du début de la transmission de la dernière rafale planifiée en omettant les intervalles de temps pendant lesquels des rafales seront transmises. La figure 1.6 montre l'intervalle vacant après la réservation du canal pour la rafale (i). Ainsi, on peut classer les mécanismes avec réservation différée en deux catégories :

- *sans remplissage de l'intervalle vacant (without void filling)* : l'intervalle vacant n'est pas utilisé pour envoyer d'autres rafales. Le mécanisme Horizon est un mécanisme sans remplissage de l'intervalle vacant. Par exemple, la figure 1.6 montre la rafale ($i + 1$) qui ne peut pas être planifiée dans l'intervalle vacant.
- *avec remplissage de l'intervalle vacant (void filling)* : pendant l'intervalle de temps vacant, la longueur d'onde peut être utilisée par d'autres rafales (si l'intervalle est assez grand pour envoyer d'autres rafales). Le mécanisme JET est un mécanisme avec remplissage de l'intervalle vacant. Par exemple, la figure 1.6 montre la rafale ($i + 1$) qui peut être planifiée dans l'intervalle vacant puisqu'on utilise une réservation différée avec remplissage de l'intervalle vacant.

En comparant les trois mécanismes, il est clair que l'avantage du mécanisme JET par rapport à Horizon et JIT est de maximiser l'utilisation des ressources et de réduire les pertes des rafales. En effet, il permet d'effectuer plusieurs réservations en même temps et d'utiliser les longueurs d'onde pendant les intervalles vacants. Par conséquent, nous utilisons le mécanisme JET pour la simulation du réseau OBS.

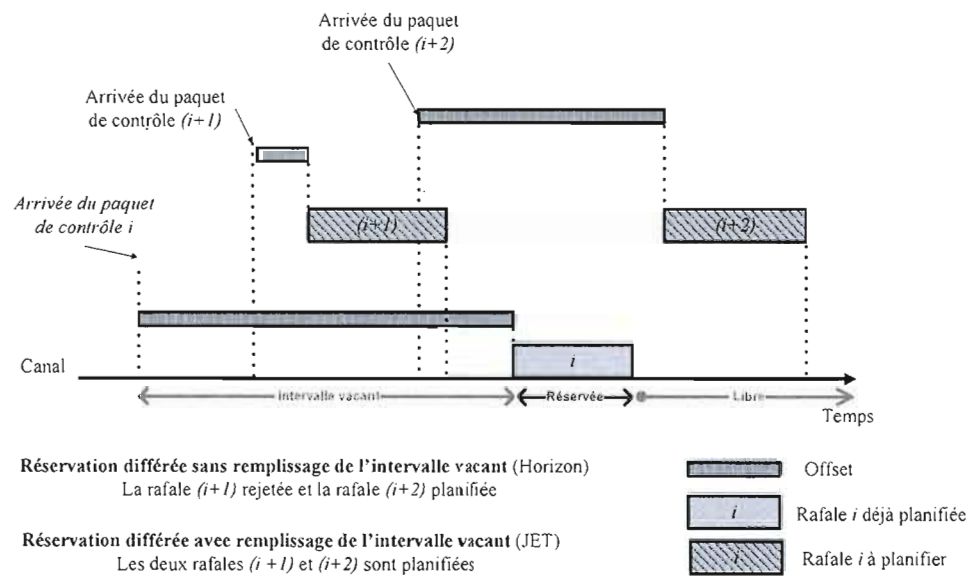


Figure 1.6 Réservation différée.

1.7.4 Contention et perte de paquets

Une rafale est rejetée dans deux cas.

Le premier cas est lorsque l'offset est insuffisant, c'est-à-dire lorsque la rafale rattrape et dépasse le paquet de contrôle. Ainsi, la rafale va être reçue par un nœud avant la réception du paquet de contrôle. Dans ce cas, la rafale est rejetée puisqu'aucune réservation n'a été effectuée.

Le deuxième cas de rejet de rafale se réalise lorsqu'il y a une contention : le paquet de contrôle est reçu par le nœud, mais il n'y a pas de place libre dans le canal pour planifier l'envoi de la rafale. Ainsi, le nœud rejette le paquet de contrôle. Lorsque la rafale arrive au nœud, elle sera rejetée puisqu'aucune réservation n'a été effectuée.

Plusieurs travaux ont été réalisés afin d'optimiser l'utilisation de la bande passante en évitant la contention (Jin et Yang, 2006; Farahmand, Zhang et Jue, 2007; Vokkarane,

Jue et Sitaraman, 2002; Jue et Vokkarane, 2005). Cinq méthodes ont été essentiellement étudiées :

1. *Routage par déflexion* : lorsqu'aucune longueur d'onde n'est disponible pour une rafale, cette dernière est envoyée sur une autre fibre (lien) contenant une longueur d'onde libre,
2. *Mise en mémoire tampon* : il est possible de mettre la rafale dans une mémoire tampon optique (*fiber delay lines* - FDLs) (Hunter, Chia et Andonovic, 1998). Cependant, il est difficile physiquement d'en créer. En effet, il faut 200 Km de fibre pour retarder une rafale pendant 1 ms (Hunter, Chia et Andonovic, 1998; Jue et Vokkarane, 2005).
3. *Conversion de longueur d'onde* : si la longueur d'onde utilisée pour une rafale n'est pas libre au niveau d'un nœud, la rafale peut être convertie et envoyée sur une autre longueur d'onde qui est libre.
4. *Segmentation des rafales* : cette technique consiste à diviser une rafale en segments. La partie de la rafale qui n'est pas en contention est envoyée normalement. L'autre partie de la rafale, qui est en contention, est envoyée sur un chemin alternatif (Detti, Eramo et Listanti, 2002; Jue et Vokkarane, 2005).
5. *Contrôle d'admission* : on peut contrôler le taux de perte dans le réseau en contrôlant le débit de burstification au niveau des nœuds d'entrée (Jin et Yang, 2006; Farahmand, Zhang et Jue, 2007; Aly, Zhani et Elbiaze, 2008; Aly, Zhani et Elbiaze, 2007a; Aly, Zhani et Elbiaze, 2007b; Aly, Zhani et Elbiaze, 2009; Zhani, Fouad et Elbiaze, 2009a).

Dans le présent travail, nous proposons un mécanisme de contrôle d'admission qui permet de contrôler le *débit de burstification des rafales*, c'est-à-dire le débit avec lequel les rafales sont injectées dans le réseau. L'objectif est de contrôler leur taux de perte dans le réseau.

1.8 Conclusions

A partir de ces travaux, nous avons déduit un ensemble de conclusions :

- La majorité des travaux s'intéressent à la prévision du débit de paquets entrants ou sortants du routeur exprimé en mégabit par seconde. Les mesures sont considérées comme les valeurs d'une série temporelle. Par conséquent, nous avons recours aux modèles de prévision des séries temporelles. Les entrées des modèles sont les anciennes observations du débit.
- Juger si le trafic d'un réseau est prédictible ou non est une tâche très difficile ; en effet, le comportement du trafic varie énormément dans le temps et dans l'espace : il dépend du type du lien ou du réseau et du moment auquel on prévoit le trafic. Ceci implique qu'un modèle de prévision doit être capable de s'adapter au trafic étudié afin de donner des prévisions assez précises à l'utilisateur.
- La complexité des modèles de prévision dépend du nombre de paramètres utilisés (cela sera discuté à la conclusion du prochain chapitre). Cependant, la complexité de l'algorithme d'apprentissage des paramètres du modèle ne pose pas un problème puisqu'il est exécuté une seule fois lors de l'identification des paramètres. Il peut être exécuté d'une façon périodique pour les mettre à jour sans bloquer le calcul de la prévision. Par exemple, le modèle prévoit la valeur future du débit chaque 100 ms, tandis que l'algorithme d'apprentissage peut être exécuté chaque 5 minutes dans un processus parallèle pour mettre à jour les paramètres.
- Chaque modèle de prévision présente des avantages et des inconvénients. Les travaux précédents ont montré que, pour effectuer une prévision à court terme (des millisecondes jusqu'aux minutes), les modèles complexes (tels que les mo-

dèles autosimilaires etc.) n'améliorent nécessairement pas la performance de la prévision en termes d'erreur. De plus, leur intégration pratique dans des équipements sera difficile à cause de la complexité de leurs calculs. D'autre part, à une faible échelle de temps (des millisecondes jusqu'aux minutes), le trafic peut être considéré stationnaire au sens large. Par conséquent, le modèle ARIMA ne donne pas une meilleure performance par rapport au modèle ARMA. Dans le présent travail, nous avons choisi d'utiliser les modèles ARMA, ARIMA, LMMSE et α -SNF vu leur simplicité en termes de complexité des calculs et leur performance en termes de précision.

- Il n'y a pas de règle générale qui permet de choisir les variables d'entrée du modèle de prévision et de déterminer le nombre de retards et la granularité des données qu'il faut considérer. L'effet de la taille *des données d'apprentissage* (c.-à-d. les données qui doivent être utilisées pour définir les paramètres du modèle) n'a pas été suffisamment étudiée.
- Généralement, les prétraitements des données, tels que le lissage et l'échantillonnage, ont une influence sur la performance de la prévision en termes d'erreur.
- L'erreur de la prévision que l'on peut tolérer dépendra du protocole qui va exploiter la valeur donnée par le modèle de prévision. Ainsi, dans le présent travail, l'objectif sera de minimiser l'erreur. La décision, quand à savoir si cette erreur est tolérable ou non, est effectuée dans un second temps lorsque la valeur est utilisée pour une application bien définie.
- L'exploitation de la prévision dans la gestion des files d'attente, la planification des capacités et la réservation des ressources, donne de bons résultats malgré l'erreur de la prévision.

Dans le prochain chapitre, nous allons définir une méthodologie de prévision détaillée. Par la suite, à l'aide d'un ensemble d'expérimentations sur des traces de trafic extraite d'un réseau réel, nous essayerons d'établir un ensemble de recommandations permettant de paramétrer le modèle de prévision, à savoir le choix des variables d'entrée, le nombre de retards, la taille des données d'apprentissage ainsi que la granularité des données.

CHAPITRE II

EFFET DES PARAMÈTRES DU MODÈLE SUR LA PERFORMANCE DE LA PRÉVISION

«En matière de prévision, le jugement est supérieur à l'intelligence. L'intelligence montre toutes les possibilités pouvant se produire. Le jugement discerne parmi ces possibilités celles qui ont le plus de chance de se réaliser»

Gustave Le Bon

Jusqu'à présent, il n'y a pas encore de règle bien définie pour paramétrer les modèles de prévision. Ainsi, il est important d'effectuer une évaluation de l'effet des paramètres des modèles sur la performance de la prévision en termes d'erreur en se basant sur des traces réelles de trafic.

Ce chapitre s'intéresse à la prévision à court terme du débit au sein d'un lien. Le débit est défini comme étant la quantité de données reçue par le routeur pendant un intervalle de temps (exprimé en Mbps) (Zhani, Elbiaze et Kamoun, 2008; Zhani, Elbiaze et Kamoun, 2009b). L'étude couvre les paramètres suivants :

- le nombre de retards (c.-à-d. anciennes observations) utilisés comme entrées du modèle de prévision.
- la quantité de données utilisée pour estimer les paramètres du modèle de prévision (appelée *données d'apprentissage*).

- la granularité du trafic c.-à-d. l'intervalle de temps qui sépare deux mesures consécutives.
- les variables exogènes qui peuvent être utilisées pour prévoir le débit du lien. Les variables exogènes sont différentes des retards (le nombre de paquets, le trafic filtré, etc.).

Toutes les expérimentations effectuées dans ce chapitre se basent sur deux traces de trafic réel provenant de deux liens ayant des caractéristiques différentes (type du lien, capacité et charge).

Dans ce qui suit, nous commençons par définir la méthodologie de la prévision qui décrit les étapes à effectuer avant d'appliquer le modèle de prévision. Ensuite, nous décrivons les traces réelles utilisées dans les expérimentations. Enfin, nous analysons les différents paramètres qui peuvent avoir une influence sur la qualité de la prévision.

2.1 Méthodologie de la prévision du trafic

2.1.1 Méthodologie

La prévision du trafic peut être appliquée à différents niveaux du modèle protocolaire d'Internet. En effet, elle peut être appliquée au niveau de la couche application, au niveau de la couche transport TCP (*Transport Control Protocol*) ou UDP (*User Datagram Protocol*), au niveau de la couche IP (*Internet Protocol*) ou au niveau des couches basses telles que la couche liaison. Le choix du niveau dépend principalement de l'application qui va exploiter la prévision.

Le présent travail s'intéresse principalement au débit de données au niveau de la couche réseau. L'avantage de travailler à ce niveau est que le trafic va contenir toutes les données provenant des couches supérieures (application et transport). *Le débit* est

la quantité de données reçue par un nœud exprimée en mégabit par intervalle de temps (Mbps). Cet intervalle est appelé *granularité*.

Le débit est vu comme étant une série temporelle (sect. 1.5). Les techniques d'analyse et de prévision du trafic seront ainsi indépendantes du type de réseau et seront appliquées aux mesures du débit au niveau de la couche IP.

La méthodologie de prévision est représentée à la figure 2.1. On note par $y(t)$ la valeur à prédire, $\hat{y}(t)$ la valeur prédite par le modèle de prévision, μ la moyenne estimée sur l'ensemble des données utilisées et n le nombre des données disponibles. Les entrées du modèle de prévision sont les variables $x_1, \dots, x_i, \dots, x_n$. Soit $x_i(t-j)$ la valeur de la variable x_i mesurée à l'instant $t-j$. Ainsi, le modèle utilise les variables $x_i(t-j)$ pour prédire $y(t)$.

Par exemple, $y(t)$ pourrait être le débit au sein d'un lien. $x_i(t-j)$ peut être une ancienne observation de $y(t)$, c'est-à-dire un retard tel que $y(t-1)$ ou $y(t-2)$ ou bien une autre variable exogène mesurée à l'instant $t-j$.

Les traces de trafic disponibles sont divisées en deux ensembles. Le premier ensemble constitue $p\%$ des données (généralement 50 %). C'est l'ensemble de données utilisé pour la *phase d'apprentissage* qui consiste à initialiser et optimiser les paramètres du modèle de prévision.

Le deuxième ensemble de données est utilisé pour la *phase de validation* qui consiste à comparer les résultats de la prévision avec les données réelles afin d'évaluer la performance de la prévision.

Ainsi, pour chaque entrée $x_i(t-j)$ mesurée à l'instant $t-j$, le modèle calcule $\hat{y}(t)$ la prévision pour $y(t)$. Par la suite, l'erreur de la prévision est calculée.

Dans ce travail, nous avons choisi les modèles ARMA et α -SNF comme modèles de prévision pour leurs simplicités en termes de complexité des calculs et leurs performances en termes de précision.

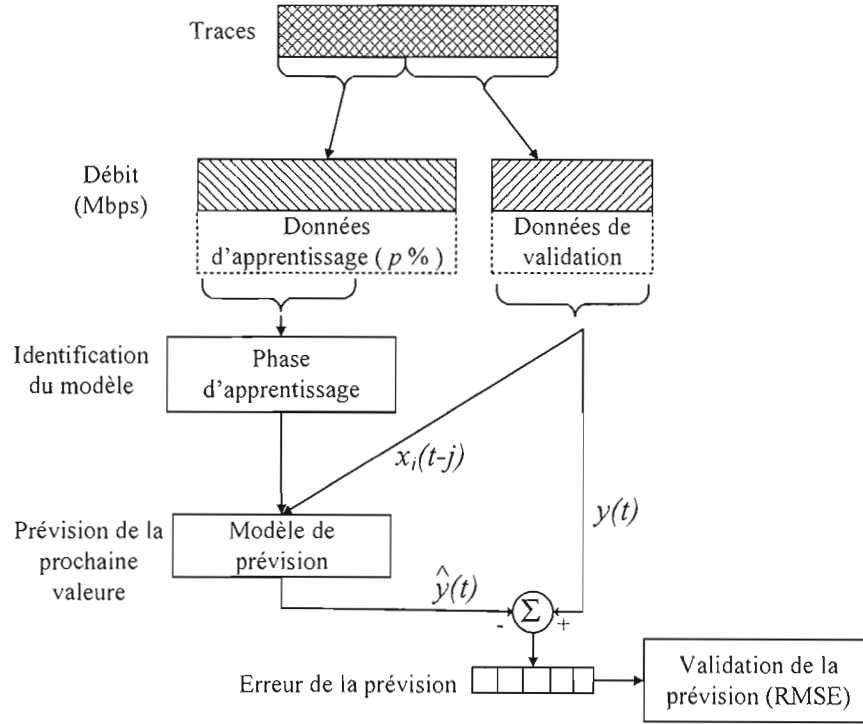


Figure 2.1 Méthodologie de la prévision.

2.1.2 Critères de performance

Pour mesurer la performance de la prévision, nous avons utilisé deux critères :

- L'erreur quadratique moyenne (*Mean Squared Error* - MSE) ou sa racine carrée (RMSE) : c'est le critère le plus simple qui mesure l'erreur de la valeur prédite par rapport à la valeur réelle. Le MSE et le RMSE sont définis respectivement par :

$$MSE = \frac{\sum_{t=1}^n [y(t) - \hat{y}(t)]^2}{n}, \quad (2.1)$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^n [y(t) - \hat{y}(t)]^2}{n}}, \quad (2.2)$$

où n est le nombre de points étudiés. Avec ce critère, une valeur atypique de la série a une grande influence sur l'erreur, ce qui peut poser des problèmes

en cas de valeur aberrante. Il est donc important, si l'on utilise ce critère, de filtrer ou de lisser les données. Ainsi ce critère pénalise les grandes erreurs : les modèles, qui se trompent souvent mais peu, seront moins pénalisés que ceux qui se trompent rarement mais beaucoup.

- La variance relative moyenne (*Average Relative Variance* - ARV) : elle est définie par :

$$ARV = \frac{\sum_{t=1}^n [y(t) - \hat{y}(t)]^2}{\sum_{t=1}^n [y(t) - \mu]^2}, \quad (2.3)$$

où n est le nombre de points étudiés et μ est la moyenne de $y(t)$ pour ces points. Ce critère constitue aussi le rapport entre le MSE et la variance des données utilisées. On remarque que si, à chaque instant t , la prévision est égale à la moyenne, l'ARV sera de 1. Si l'ARV est inférieur à 1 alors le modèle de prévision offre un gain par rapport à l'utilisation de la moyenne comme valeur prédite. Contrairement au MSE, le critère ARV ne dépend pas de l'échelle ou de l'unité utilisée ; il ne change pas arbitrairement avec l'augmentation du nombre des données et il donne à l'utilisateur une idée claire sur la qualité de l'approximation.

2.2 Analyse des traces d'un réseau réel

Dans cette section, nous présentons les traces de trafic qui sont utilisées pour définir le modèle de prévision et pour effectuer les d'expérimentations. De plus, nous présentons les différents prétraitements effectués sur les données. Nous analysons aussi les différentes caractéristiques du trafic qui pourront avoir un effet sur la performance de la prévision.

Dans ce qui suit, on note par $y(t, bps)$ et $y(t, pps)$ le débit à l'instant t exprimé respectivement en Mbps ou en nombre de paquets par seconde (pps). Afin de simplifier, la variable $y(t, bps)$ est aussi notée $y(t)$.

2.2.1 Présentation des traces et prétraitements

Le premier ensemble de données « *Auckland-VIII*¹ » est un ensemble de traces collectées sur un lien Ethernet de capacité 100 Mbps qui relie l'université d'*Auckland* à Internet. Ce sont les traces de deux semaines contenant les en-têtes de paquets IP capturés par une carte *Endace DAG3.5E TAP Ethernet* en décembre 2003.

Le deuxième ensemble de données est l'ensemble « *CESCA-I* ». Ce sont des traces de trois heures d'en-têtes IP capturés sur un lien d'un Gbps avec une carte *Endace DAG4.2GE dual Gigabit Ethernet* en février 2004 dans le réseau *the Anella Cientifica, the Catalan R&D network*.

Nous avons analysé les traces avec l'outil *libtrace*² qui permet d'extraire le débit exprimé en *bits par seconde* (bps) ou en *paquets par seconde* (pps), le nombre de connexions TCP ainsi que le temps d'aller-retour. L'analyse est effectuée au niveau des paquets IP.

Nous avons effectué les mêmes analyses sur les données collectées à différents moments de la journée; les résultats trouvés sont similaires. Ainsi, nous présentons seulement les résultats trouvés en utilisant 60 minutes de traces mesurées à dix heures du matin. Pour la prévision, nous avons utilisé 50 % des données (30 mn) pour la phase d'apprentissage et les 50 % restants (30 mn) pour la phase de validation du modèle de prévision.

1. Les données sont disponibles via le site *the National Science Foundation and the NLANR Measurement and Network Analysis Group* (<http://pma.nlanr.net/Special/>)

2. *Wand Network Research Group*, The *libtrace* trace-processing library.
<http://research.wand.net.nz/software/libtrace.php>

2.2.2 Analyse des traces

La figure 2.2 montre les durées des différentes connexions TCP qui traversent le lien Auckland. Plus de 80 % des connexions ne durent pas plus d'une seconde. La durée moyenne des connexions est de 8,23 s. La connexion la plus longue a duré 36 minutes.

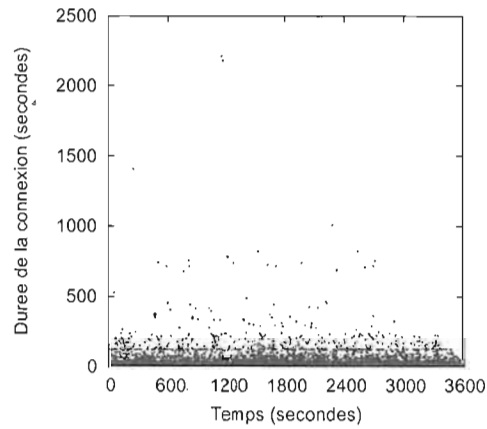


Figure 2.2 Durée des connexions TCP (Auckland-VIII).

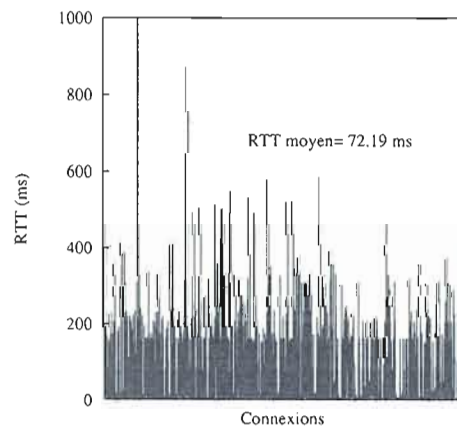


Figure 2.3 RTT des connexions TCP (Auckland-VIII).

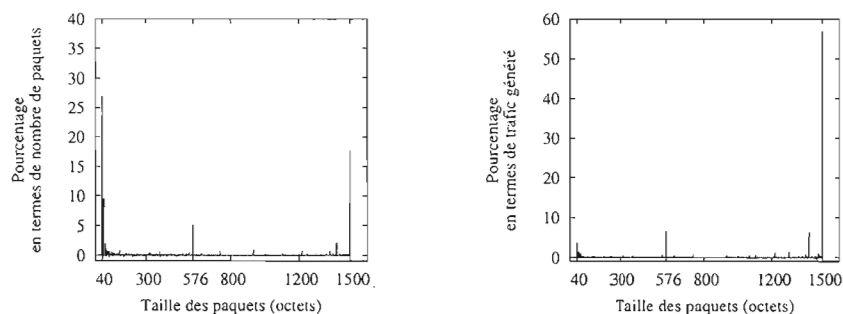
La figure 2.3 montre le RTT pour toutes les connexions TCP. Plus de 70 % des connexions ont un RTT entre 72 ms et 100 ms. Le RTT moyen et la durée moyenne des connexions peuvent être intéressantes pour identifier la granularité avec laquelle il faut analyser et prévoir les données.

Une comparaison entre les traces Auckland-VIII et CESCA-I est présentée dans le tableau 2.1. Elle montre les différences entre les caractéristiques des deux traces pour la même durée (une heure). Le lien Auckland de 100 Mbps a beaucoup moins de charge que le lien CESCA de 1 Gbps en termes de nombre de paquets ou en termes de trafic (mesurée en Mbps). On remarque aussi l'importante variance des données CESCA-I ; cela pourrait dégrader la précision de la prévision. Cependant, la durée moyenne d'une connexion est la même pour les deux traces (≈ 8 s).

Tableau 2.1 Caractéristiques des traces de trafic Auckland-VIII et CESCA-I

Statistiques	Lien Auckland-VIII	Lien CESCA-I
Capacité du lien	100 Mbps	1 Gbps
Durée (heure)	1	1
Taille des traces	60 MB	20,5 GB
Débit moyen (Mbps)	2,83	487,16
Variance du débit (Mbps)	1,280	797,781
Débit moyen (pps)	972,73	100 036,14
Durée moyenne d'une connexion (seconde)	8,23	8,14

La figure 2.4(a) montre la distribution des tailles des paquets par rapport au nombre de paquets (trafic Auckland-VIII). La plus petite taille, 40 octets, représente 27 % du nombre des paquets. Ils sont principalement des paquets TCP avec des drapeaux ACK, SYN, FIN, or RST. Les paquets de taille 1500 octets constituent 17,67 % du nombre total de paquets. Cette taille correspond à la taille maximale d'un paquet IP. On trouve aussi beaucoup de paquet de taille 1420 octets (2,10 %). On remarque aussi la présence d'un nombre important de paquets de taille 576 octets, ce qui reflète



(a) Distribution considérant le nombre de paquets.

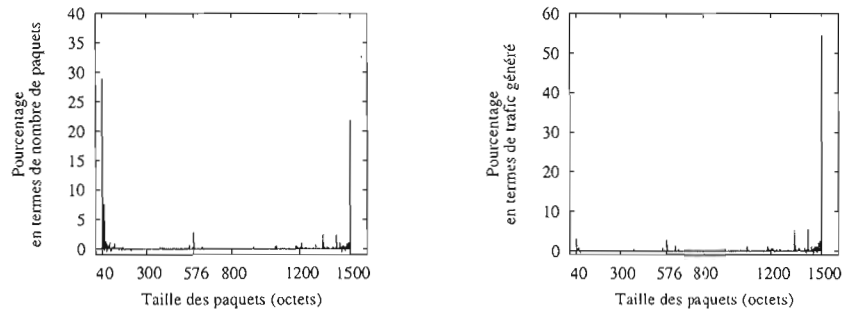
(b) Distribution considérant le trafic généré.

Figure 2.4 Distribution des tailles des paquets pour le trafic Auckland.

l'implémentation de TCP sans « *Path MTU discovery* », qui utilise la taille 536 octets (plus 40 octets d'en-tête) comme la valeur par défaut de la taille maximale du segment TCP (RFC 879) (Postel, 1983).

La figure 2.4(b) montre la distribution des tailles de paquets par rapport à la quantité de trafic générée. On définit le trafic généré comme la quantité de données exprimée en mégabit. La figure 2.4(b) montre que les paquets de taille 1500 octets constituent 57 % du trafic généré. Cependant, ils constituent seulement 17,67 % du nombre total de paquets (fig. 2.4(a)). Par contre, bien que 27 % des paquets soient de faibles tailles, ils génèrent moins de 5 % du trafic. Cette observation est conforme aux résultats trouvés par (Shao et Trajkovic, 2004) qui ont découvert que le trafic est bimodal : la majorité du trafic est transportée par un faible nombre de paquets et la majorité des paquets transporte une quantité plus faible du trafic. Les mêmes observations peuvent être remarquées pour le cas du trafic CESCA (fig. 2.5).

Ainsi, le comportement du trafic est influencé par celui des paquets de grandes tailles. Le filtrage du trafic peut être fait selon la taille des paquets et la série résultante peut être utilisée comme entrées du modèle de prévision. Ce point va être développé davantage lorsque les variables exogènes seront utilisées comme entrées du modèle.



(a) Distribution considérant le nombre de paquets.

(b) Distribution considérant le trafic généré.

Figure 2.5 Distribution des tailles des paquets pour le trafic CESCA.

En tenant compte de ces caractéristiques, nous analysons, dans ce qui suit, les différents paramètres qui ont un impact sur la qualité de la prévision en termes d'erreur.

2.3 Analyse des différents paramètres du modèle de prévision

2.3.1 Choix des variables d'entrée du système

Dans ce qui suit, nous étudions comment choisir les variables d'entrée du modèle de prévision. Les variables candidates pour être les entrées du modèle sont, généralement, les retards $y(t - i)$ et les variables exogènes $x(t - i)$.

Pour choisir parmi ces variables, nous allons considérer deux métriques : l'information mutuelle et le coefficient de corrélation. Ces deux métriques sont calculées en utilisant la variable candidate pour l'entrée du système et la variable à prédire.

Information mutuelle

L'entropie : elle mesure l'incertitude de l'information formulée en termes de la théorie des probabilités (Kantz et Schreiber, 2006).

$$H(X) = - \sum_x P(x) \log_2 P(x), \quad (2.4)$$

où $P(x) = P[X = x]$. Intuitivement, l'entropie mesure le degré de « surprise » qu'on doit sentir suite à la connaissance des observations des variables aléatoires. En effet, quand $P(x) \rightarrow 0$, nous serons surpris de voir l'évènement x se produire, donc on a $H \rightarrow \log_2 N$ où N est le nombre d'états possibles de X . Si $p(x) \rightarrow 1$, on ne sera pas surpris si x se produit, donc $H \rightarrow 0$.

L'entropie conjointe $H(X, Y)$ de deux variables aléatoires X et Y est définie par :

$$H(X, Y) = \sum_{x, y} P(x, y) \log_2 \frac{1}{P(x, y)} \quad (2.5)$$

où $P(x, y) = P[X = x, Y = y]$ la probabilité conjointe de x et y .

L'information mutuelle : elle mesure la robustesse de la relation (dans le sens probabiliste) et la dépendance mutuelle entre les valeurs des variables X et Y (Shannon, 1948). Soit $I(X, Y)$ l'information mutuelle de deux variables aléatoires X et Y . Elle est définie par :

$$I(X, Y) = H(X) + H(Y) - H(X, Y). \quad (2.6)$$

On peut aussi utiliser un coefficient appelé « *coefficient de dépendance* » basé sur la mesure de l'information mutuelle entre deux variables :

$$U(X, Y) = 2 \times \left[\frac{H(X) + H(Y) - H(X, Y)}{H(X) + H(Y)} \right]. \quad (2.7)$$

Si les deux variables sont indépendantes alors $H(X, Y) = H(X) + H(Y)$, donc $U(X, Y) = 0$.

Si les deux variables sont complètement dépendantes alors on a :

$$H(X, Y) = H(X) = H(Y), \quad (2.8)$$

donc $U(X, Y) = 1$. Ainsi, $U(X, Y)$ est similaire au coefficient de corrélation entre les deux variables pour des dépendances non linéaires.

L'utilisation de l'information mutuelle est un moyen assez efficace pour exprimer la pertinence des variables en entrée pour la prédiction d'une variable en sortie ; En effet, lorsque celle-ci est élevée, il y a une forte corrélation entre la variable d'entrée du modèle et sa sortie.

Coefficient de corrélation

Dans la théorie des probabilités et des statistiques, le coefficient de corrélation indique la force et la direction d'un rapport linéaire entre deux variables aléatoires. Plusieurs coefficients qui mesurent le degré de corrélation ont été proposés selon la nature des données. Le coefficient de corrélation le plus connu est le coefficient *Pearson product-moment* (Cohen, 1988). Il est obtenu par la division de la covariance des deux variables par leurs écart-types. Soient X et Y deux variables aléatoires ayant respectivement les espérances mathématiques μ_X et μ_Y et les écart-types σ_X et σ_Y . La corrélation $\rho_{X,Y}$ entre X et Y est définie par :

$$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X - \mu_X)(Y - \mu_Y))}{\sigma_X \sigma_Y}. \quad (2.9)$$

Sachant que $\mu_X = E(X)$, $\sigma_X^2 = E(X^2) - E^2(X)$ et de même pour Y . On peut aussi écrire :

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\sqrt{E(Y^2) - E^2(Y)}} \quad (2.10)$$

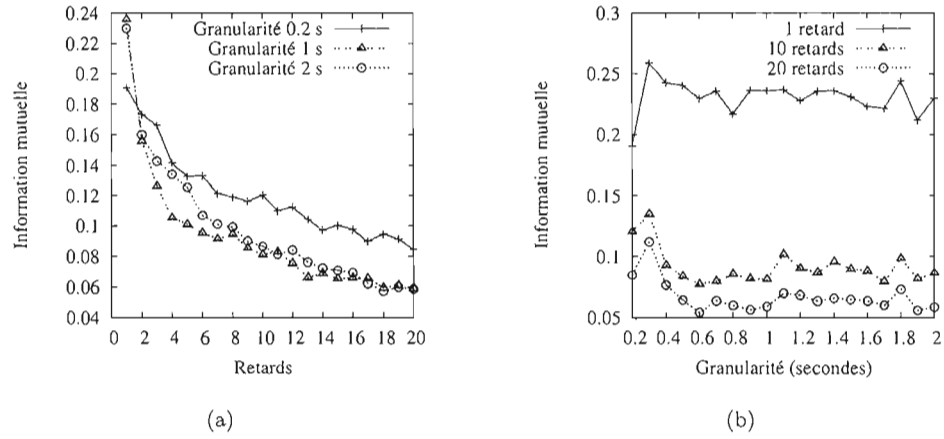


Figure 2.6 Information mutuelle par rapport au nombre de retards et à la granularité (données Auckland-VIII).

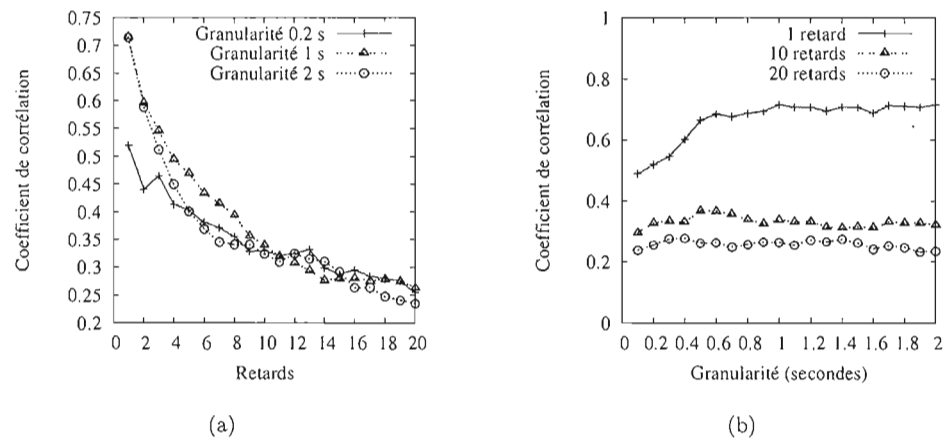
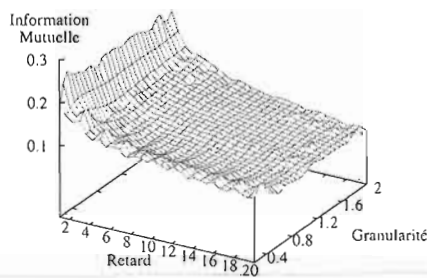


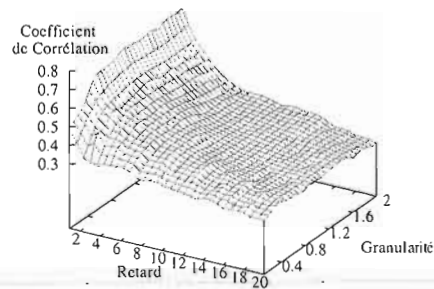
Figure 2.7 Coefficient de corrélation par rapport au nombre de retards et à la granularité (données Auckland-VIII).

La figure 2.6(a) montre l'information mutuelle des retards $y(t - i)$ (i varie de 1 jusqu'à 20) avec $y(t)$ pour quelques granularités (0, 2 s, 1 s et 2 s). Elle montre que l'information mutuelle diminue lorsque des retards plus anciens (i croît) sont considérés. Le coefficient de corrélation représenté dans la figure 2.7(a) confirme cette observation.

Les figures 2.6(b) et 2.7(b) montrent l'effet de la granularité sur la corrélation de $y(t)$ avec les différents retards ($t - 1$, $t - 10$ et $t - 20$). Elles montrent que pour les deux métriques utilisées, l'information mutuelle et la corrélation ne changent pratiquement pas lorsqu'on augmente la granularité. Ainsi, la corrélation entre $y(t)$ et un retard $y(t - i)$ est pratiquement la même quelle que soit la granularité.



(a)



(b)

Figure 2.8 (a) Information mutuelle et (b) coefficient de corrélation pour différentes granularités (données Auckland-VIII).

La figure 2.8 montre l'information mutuelle et le coefficient de corrélation pour toutes les granularités variant de 0,1 s à 2 s et pour tous les retards variant de 1 à 20. Elle montre que la corrélation des retards $y(t - i)$ avec la valeur à prédire $y(t)$ décroît lorsqu'on utilise des anciens retards quelle que soit la granularité. D'après les deux métriques, pour une granularité fixée, le retard $y(t - 1)$ est le plus corrélé à $y(t)$. Cela nous suggère que c'est la variable la plus importante à utiliser comme entrée du modèle de prévision.

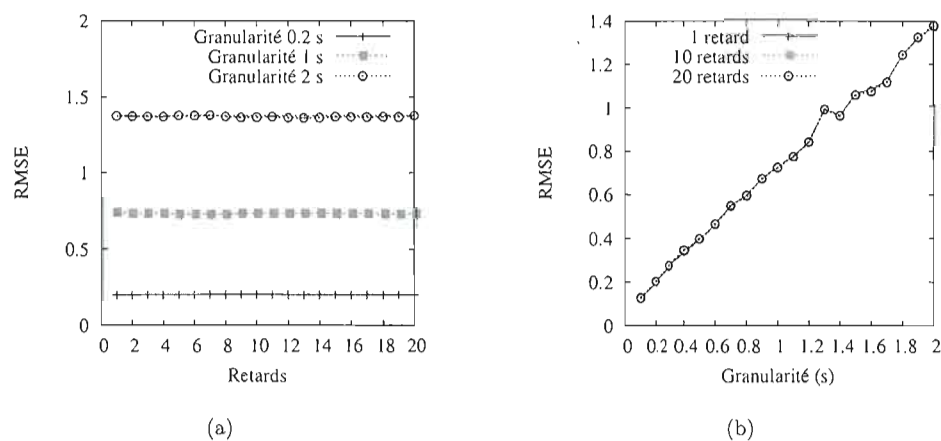


Figure 2.9 Erreur de la prévision par rapport au nombre de retards et à la granularité (modèle ARMA, données Auckland-VIII).

Afin de valider cette observation, nous avons effectué plusieurs prévisions en utilisant, comme entrées, des retards variant de 1 à 20. Nous avons varié les granularités de 0,1 s à 2 s. Ainsi, chaque point dans les figures 2.9 ci-dessous représente l'erreur de la prévision de 30 minutes obtenue avec le modèle de prévision utilisant i retards et appliqué sur des données de granularité fixe.

La figure 2.9(a) montre l'erreur de la prévision (RMSE) par rapport au nombre de retards. Elle montre que l'augmentation du nombre de retards, qui sont utilisés comme entrées du modèle, n'améliore pas la performance de la prévision en termes d'erreur. La figure 2.9(b) montre l'erreur de la prévision (RMSE) par rapport à la granularité. Il est clair que la performance se dégrade lorsque la granularité augmente.

La figure 2.10 montre la performance de la prévision en termes d'erreur (RMSE) obtenue avec le modèle ARMA en utilisant de 1 à 20 retards et des granularités variant de 0,1 s à 2 s. Il est clair que, quelle que soit la granularité, l'utilisation de plus d'un retard comme entrée n'améliore pas la performance de la prévision. De plus, la précision de la prévision se dégrade lorsque la granularité augmente.

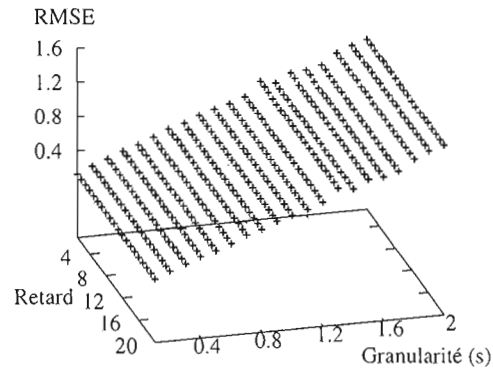
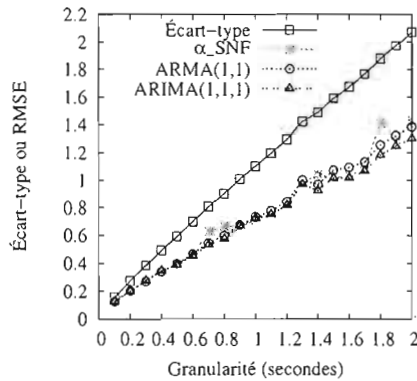


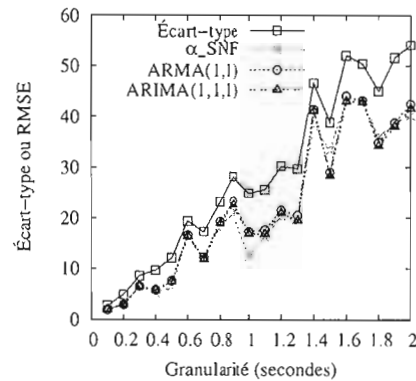
Figure 2.10 Erreur de la prévision (modèle ARMA, données Auckland-VIII).

2.3.2 Choix de la granularité du trafic

La granularité est un paramètre important puisqu'elle a un effet sur la précision de la prévision.



(a) Données Auckland-VIII.



(b) Données CESCA-I.

Figure 2.11 L'écart-type et l'erreur de prévision (RMSE) par rapport à la granularité.

Nous avons effectué un ensemble de prévisions en utilisant des données avec des

granularités variant de 100 ms à 2000 ms. Nous considérons les granularités qui sont multiples du RTT (≈ 100 ms).

La figure 2.11 montre l'erreur de prévision RMSE obtenue pour des prévisions effectuées avec des données de différentes granularités ainsi que l'écart-type des données (c.-à-d. la racine carrée de la variance). Elle montre que l'erreur de la prévision (RMSE) augmente quand la granularité des données augmente. Par conséquent, la granularité doit être choisie en fonction de l'erreur RMSE qui peut être tolérée par l'application utilisant la prévision.

Par exemple, pour améliorer le protocole TCP, nous avons besoin de la prévision du débit pour un ou plusieurs RTT. Les résultats expérimentaux montrent que nous pouvons effectuer une prévision pour un horizon prévisionnel de 600 ms ($\approx 6 \times RTT$) avec une erreur (RMSE) de 0,5 Mb pour le trafic d'Auckland et de 15 Mb pour le trafic CESCO. Ainsi, l'administrateur peut juger si cette erreur est tolérable ou non en fonction de l'utilisation ultérieure de la valeur prédite.

La figure 2.11 montre aussi l'écart-type des données par rapport à la granularité des deux traces. Nous remarquons que, lorsque la granularité augmente, l'écart-type augmente et que l'erreur augmente de la même manière. En effet, quand la granularité augmente, la variance des données augmente. Par conséquent, le trafic devient moins prévisible (RMSE augmente).

La figure 2.11 montre aussi que les erreurs de prévision obtenues avec les différents modèles utilisés (ARMA, ARIMA ou α -SNF) sont pratiquement les mêmes quelle que soit la granularité. Ainsi, ces modèles offrent quasiment la même performance de prévision en termes d'erreur.

2.3.3 Choix de la taille des données d'apprentissage

Les modèles de prévision utilisés nécessitent une phase d'apprentissage, c'est-à-dire une phase pendant laquelle les paramètres du modèle sont estimés et optimisés afin

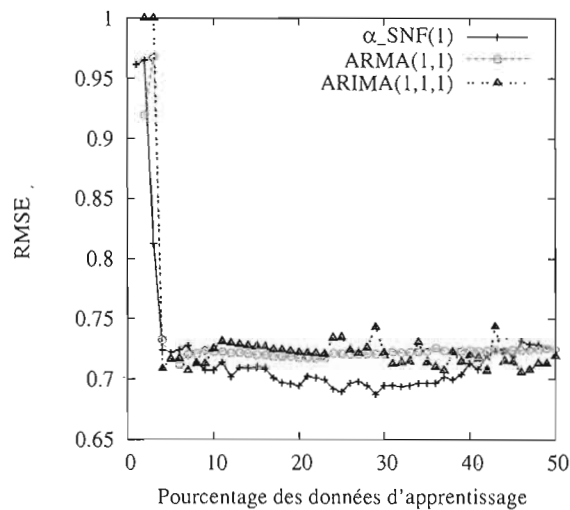


Figure 2.12 Erreur de la prévision par rapport à la taille des données d'apprentissage (données Auckland-VIII, granularité 1 s).

de donner la meilleure performance de prévision en termes d'erreur. Ainsi, on s'intéresse à la taille des données d'apprentissage utilisées pour cette phase.

Nous avons effectué plusieurs expériences en variant le pourcentage utilisé pour les données d'apprentissage (p %) de 10 à 50 %. Nous considérons les données de granularité égale à une seconde. À chaque expérience, nous utilisons p % des données disponibles pour prévoir les derniers 50 %. A priori, on peut penser qu'élargir la taille des données d'apprentissage pourrait améliorer la performance de la prévision puisque cela donne plus d'informations sur les caractéristiques des données et par la suite, les paramètres du modèle pourraient être plus précis.

Contrairement à ce que l'on peut penser, augmenter la taille des données d'apprentissage n'améliore nécessairement pas l'erreur de prévision selon les résultats des expérimentations. La figure 2.12 montre l'erreur de la prévision (RMSE) en fonction du pourcentage des données d'apprentissage utilisées. Pour le modèle ARMA, le RMSE est quasiment stable à partir de 5 %. Pour le modèle ARIMA, le RMSE augmente à partir

de 8 %. Pour le modèle α -SNF, lorsque la taille des données d'apprentissage augmente, l'erreur de la prévision diminue jusqu'à un certain point (25 %) après lequel l'erreur augmente.

La figure montre que si la taille des données d'apprentissage est faible (< 5 %), l'erreur de la prévision devient très importante. En effet, dans ce cas, les données d'apprentissage ne sont pas suffisantes pour estimer avec précision les paramètres du modèle.

Si on augmente la taille des données d'apprentissage, il y a deux cas possibles. Le premier cas est quand le modèle ne dispose pas d'assez de paramètres pour apprendre les données correctement, ainsi, il n'est pas capable de capturer toutes les caractéristiques des données. Par conséquent, l'augmentation de la taille des données d'apprentissage n'améliorera pas la précision vu que le modèle est limité par son faible nombre de paramètres. Les résultats des sections précédentes ont montré que, dans notre cas, même l'augmentation du nombre de paramètres (retards) n'améliore pas la performance du modèle.

Le deuxième cas est quand le modèle dispose de trop de paramètres et les utilise pour apprendre toutes les particularités des données, c'est ce que l'on appelle le *sur-apprentissage*. Dans ce cas, le modèle va capturer beaucoup de caractéristiques qui ne seront plus valides dans le futur. Ainsi, il n'est plus capable de prévoir les données puisque les particularités des données changent rapidement au cours du temps. Par conséquent, l'augmentation de la taille des données réduit la précision de la prévision.

Par conséquent, la taille des données d'apprentissage doit être suffisante pour estimer et optimiser les paramètres, mais pas très grande pour éviter que le modèle capture les particularités des données d'apprentissage. Ces résultats ont été confirmés par d'autres travaux récents (Bermolen et Rossi, 2009).

2.3.4 Pr vision du trafic en utilisant des variables exog nes

Vu la taille importante du trafic, les mesures sont limit es   quelques statistiques (telles que la taille des paquets, leur r partition ou leur nombre, le nombre de flux etc.) sur les donn es ou sur un sous-ensemble de ces donn es. Par cons quent, il serait int ressant d'avoir un mod le de pr vision capable d'avoir, comme entr es, des variables exog nes, c'est- -dire diff rentes des anciennes observations du d bit et qui permettent de pr voir le d bit.

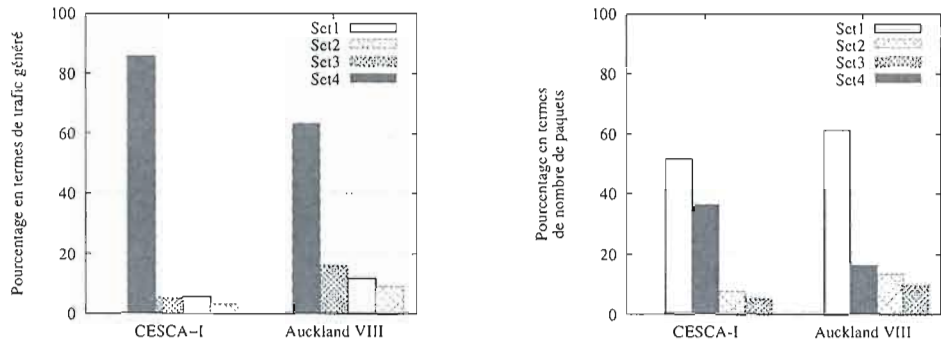
On note $y(t, bps)$ et $y(t, pps)$ le d bit total du lien exprim  respectivement en Mbps et en nombre de paquets (pps) mesur    un instant t . La granularit  consid r e est une seconde c'est- -dire $y(t, bps)$ est la mesure de la quantit  du trafic qui a travers  le lien pendant une seconde. Ainsi, nous utilisons du trafic filtr  sur la base de la taille des paquets pour pr dire le d bit total du lien $y(t, bps)$.

L'analyse du trafic (sect. 2.2) a montr  que la majorit  du trafic est port e par les paquets de grande taille (principalement 1500 octets) qui constituent un faible pourcentage en termes de nombre de paquets.   partir de la distribution de la taille des paquets (fig. 2.4 et 2.5), nous avons class  les paquets en quatre ensembles :

- Set1 : ensemble des paquets dont la taille appartient   $[0, 100]$ octets [
- Set2 : ensemble des paquets dont la taille appartient   $[100, 500]$ octets [
- Set3 : ensemble des paquets dont la taille appartient   $[500, 800]$ octets [
- Set4 : ensemble des paquets dont la taille appartient   $[800, 1500]$ octets]

Ainsi, on note par $set_i(t-1, bps)$ et $set_i(t-1, pps)$ les d bits des paquets appartenant   l'ensemble Set_i   l'instant $t-1$ exprim  respectivement en Mbps ou en nombre de paquets par seconde (pps). Ces variables exog nes peuvent  tre utilis es comme entr es du mod le de pr vision.

L'analyse du trafic montre que pour le trafic CESCA, les grands paquets (Set4) constituent 85 % du trafic en Mb, mais seulement 35 % du nombre total de paquets (fig. 2.13). Les grands paquets (Set4) dans le lien Auckland constituent seulement 16 %



(a) Répartition du trafic selon l'accumulation du trafic généré en Mbit.

(b) Répartition du trafic selon le nombre de paquets.

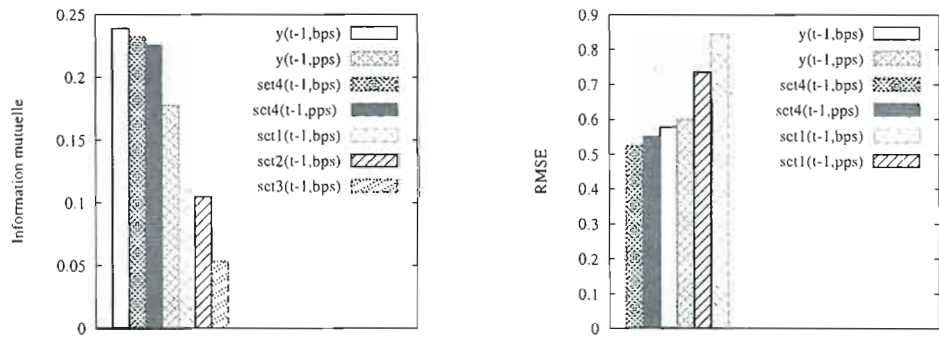
Figure 2.13 Pourcentage des différents ensembles par rapport au trafic total.

des paquets mais plus de 60 % du trafic exprimé en Mb.

On note aussi que, bien que plus de 60 % des paquets appartiennent à l'ensemble Set1 (taille ≤ 100 octets), ils constituent moins de 15 % du trafic en Mb pour le lien Auckland et moins de 8 % pour le lien CESCA. Cette observation concorde avec les résultats de (Shao et Trajkovic, 2004) qui affirment que le trafic est bimodal : la majorité du trafic est transporté par un petit nombre de paquets et la majorité des paquets transporte un petit nombre d'octets.

La figure 2.14(a) montre l'information mutuelle (éq. 2.7) entre le débit $y(t, bps)$ et chaque variable exogène. Elle montre que le débit est très corrélé avec $set4(t)$ considéré en nombre de paquet ou en trafic généré (Mbps). Ceci nous suggère d'utiliser ces variables comme entrées du modèle de prévision. La figure montre aussi que $set1(t)$, $set2(t)$ et $set3(t)$ sont moins corrélés avec le débit $y(t, bps)$.

Pour valider cette observation, le modèle α -SNF a été appliqué en utilisant ces variables exogènes. Nous avons utilisé la même méthodologie (fig. 2.1) mais en remplaçant $x(t - j)$ par la valeur de la variable exogène mesurée à l'instant $t - j$. L'objectif est de prévoir le débit $y(t, bps)$ à l'instant t .



(a) Information mutuelle entre $y(t, bps)$ et les variables exogènes.

(b) Erreur de prévision en utilisant les variables exogènes comme entrées (modèle α -SNF).

Figure 2.14 Résultats expérimentaux en utilisant les variables exogènes (données Auckland-VIII).

La figure 2.14(b) montre l'erreur de la prévision (RMSE) obtenue pour la prévision du débit en utilisant comme entrée $y(t-1, bps)$, $y(t-1, pps)$ ou bien le trafic appartenant aux ensembles $Set4(t)$ et $Set1(t)$ exprimés en nombre de paquets ou en trafic généré en Mbps à l'instant $(t-1)$.

Il est clair qu'en utilisant le nombre de paquet du Set4 (16 % du nombre total des paquets), comme variable entrée du système, on obtient la meilleure performance en termes d'erreur de prévision. La même performance est atteinte quand on utilise le trafic généré de l'ensemble Set4 comme entrée du modèle.

L'utilisation du Set1 (les paquets dont la taille ≤ 100 octets) exprimé en nombre de paquets ou en trafic généré n'améliore pas la qualité de la prévision bien qu'il constitue plus de 50 % du nombre de paquets.

Par conséquent, on peut caractériser le trafic en utilisant seulement les paquets dont la taille est supérieure à 800 octets. Les résultats montrent que les données filtrées sur la base de la taille du trafic peuvent être utilisées efficacement pour prévoir le trafic.

2.4 Conclusion

Dans ce chapitre, nous avons présenté une analyse de l'effet des différents paramètres des modèles de prévision en se basant sur des expérimentations sur des traces réelles. Cette étude a permis de déduire un ensemble de conclusions et de recommandations :

- la performance de la prévision est liée à la variance du trafic. Ainsi, plus la variance des données est grande, plus l'erreur de la prévision augmente.
- l'augmentation de la granularité dégrade la performance de la prévision. En effet, la variance des données augmente avec l'augmentation de la granularité, ce qui dégrade la performance de la prévision.
- les entrées du modèle peuvent être limitées à un retard quelle que soit la granularité des données utilisées. Les expérimentations montrent que l'augmentation du nombre de retards n'améliore pas la performance de la prévision. Cela montre que, quel que soit le modèle de prévision, la complexité des calculs ne sera pas importante.
- la corrélation de la valeur actuelle du débit avec un retard ne varie pas lorsque l'on change de granularité.
- nous avons trouvé que l'augmentation de la taille de la quantité de données pour l'apprentissage n'améliore pas la performance de la prévision en termes d'erreur. Cela permet entre autres de réduire la complexité de la phase d'apprentissage.
- le nombre de paquets ou la quantité du trafic filtrée sur la base de la taille des paquets, exprimés en pps ou Mbps, peuvent caractériser le trafic et être

utilisées comme entrées du modèle de prévision. Les résultats montrent que le comportement du trafic dépend principalement de celui des grands paquets (de tailles supérieures à 800 octets). Ainsi, la performance de la prévision est améliorée lorsque le débit des grands paquets (exprimé en pps ou Mbps) est utilisé comme entrée du modèle.

Certes, cette étude a montré les effets des différents facteurs permettant de paramétrer le modèle de prévision. Cependant, les techniques de prétraitements, notamment les techniques d'échantillonnage, peuvent aussi améliorer la prévision du trafic. Dans le chapitre suivant, nous présenterons une nouvelle technique d'échantillonnage capable d'améliorer la performance de la prévision.

CHAPITRE III

PROPOSITION D'UNE NOUVELLE TECHNIQUE D'ÉCHANTILLONNAGE

«Pour réussir, il ne suffit pas de prévoir. Il faut aussi savoir improviser»

Isaac Asimov

Nous avons étudié, dans le chapitre précédent, les effets des paramètres du modèle de prévision ainsi que l'influence de la granularité des données. Ce chapitre s'intéresse aux techniques d'échantillonnage et à leurs effets sur la prévision du trafic.

L'échantillonnage est appliqué aux valeurs de la série temporelle représentant le débit. Le débit est défini comme étant la quantité de données reçues par le routeur par intervalle de temps (exprimée en Mbps).

Lors de l'échantillonnage du trafic, il y a toujours une perte d'information. Cependant, celui-ci doit aider à mettre en valeur certaines caractéristiques selon l'utilisation ultérieure des données.

Dans notre cas, les données échantillonnées sont utilisées pour la prévision du trafic. Ainsi, idéalement, elles doivent refléter son auto-similarité et sa dépendance à long terme. De plus, les données échantillonnées serviront à prévoir le débit à l'avenir. Cela permet de prendre des décisions pour l'approvisionnement des ressources en déterminant la quantité de ressources requise pour contenir le trafic. Ainsi, il est important que les données échantillonnées gardent les variations du trafic, notamment les pics qui représentent éventuellement les congestions du lien.

Afin d'atteindre ce but, nous proposons une nouvelle technique d'échantillonnage qui préserve la structure du trafic en termes d'auto-similarité et de dépendance à long terme. En effet, nous prouvons analytiquement qu'elle préserve l'auto-similarité et la dépendance à long terme. Les expérimentations effectuées sur un trafic réel confirment ce résultat et montrent aussi que la technique proposée améliore la performance du modèle de prévision.

Dans ce qui suit, nous commençons par présenter les objectifs de l'échantillonnage des données utilisées pour la prévision du trafic. Ensuite, nous présentons la technique d'échantillonnage proposée et nous démontrons qu'elle préserve l'auto-similarité du trafic. Finalement, nous validons ces résultats par des comparaisons avec d'autres techniques d'échantillonnage et d'autres expérimentations sur la prévision.

3.1 Objectifs de l'échantillonnage

Dans ce qui suit, nous présentons les caractéristiques les plus importantes à préserver lors de l'échantillonnage si les données sont destinées à être utilisées pour la prévision du trafic.

– L'auto-similarité et la dépendance à long terme

Le trafic Internet est auto-similaire et il présente une dépendance à long terme (c.-à-d. une auto-corrélation importante) (sect. 1.4.2). Ainsi, il est important que la technique d'échantillonnage proposée préserve ces deux caractéristiques. En effet, les données échantillonnées doivent préserver l'auto-corrélation que le modèle de prévision doit capturer afin d'améliorer la performance en termes d'erreur. De plus, si les données perdent leur auto-similarité, l'augmentation de l'échelle de temps (la granularité) aura comme conséquence la perte de toutes les variations (Leland et al., 1994).

Le paramètre de *Hurst* reflète le degré d'auto-similarité et de dépendance à long terme du trafic. Ainsi, pour tester si la technique d'échantillonnage a préservé le degré d'auto-similarité du trafic original, on évalue les valeurs du paramètre de *Hurst* pour

les données originales et les données échantillonnées afin de les comparer.

– Les pics du trafic

La sporadicité du trafic et son comportement en rafales créent beaucoup de problèmes au niveau de la gestion du trafic (réservation des ressources) et au niveau des mémoires tampons (*buffer*) dans les routeurs. Ces rafales sont des pics de débit qui représentent les éventuelles congestions au niveau du lien. Ainsi, il est important que les données échantillonnées gardent les pics du trafic afin de les prévoir. Cela permettra de réserver les ressources futures ou gérer les files d'attente avec plus de précision en tenant compte des hausses du trafic.

– La précision de la prévision

Dans notre cas, la précision de la prévision, effectuée avec le trafic échantillonné, est un critère important pour le choix de la technique d'échantillonnage. Puisque notre objectif final est la prévision des ressources disponibles dans le futur, l'échantillonnage doit aider à améliorer la performance de la prévision.

Dans ce qui suit, nous présentons la technique d'échantillonnage proposée qui permet de préserver l'auto-similarité et la dépendance à long terme du trafic. Elle permet aussi d'améliorer la performance de la prévision.

3.2 Présentation de la technique d'échantillonnage basé sur le maximum (MBS)

Dans ce travail, nous utilisons le débit du trafic exprimé en Mbps qui peut être facilement obtenu par un outil de mesure du trafic.

Nous proposons de prévoir le trafic au niveau des pics au lieu du trafic original. En d'autres termes, nous visons à prévoir les pics du trafic qui représentent les états de congestions potentiels du lien. En les prévoyant, le contrôleur du réseau peut approvisionner les ressources nécessaires.

Afin d'atteindre cet objectif, nous proposons une technique d'échantillonnage baptisée

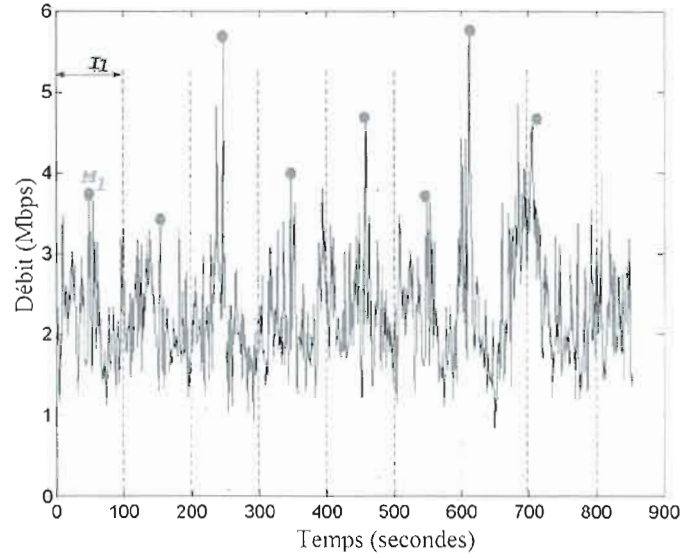


Figure 3.1 Illustration de la technique MBS.

tisée échantillonnage basé sur le maximum (*Maximum-Based Sampling* - MBS) qui prend une seule mesure durant un intervalle d'échantillonnage de taille I . Cette mesure correspond à la valeur maximale rencontrée durant l'intervalle d'échantillonnage c.-à-d. le pic du débit (fig. 3.1). Soit $x(t)$ une série temporelle. La série échantillonnée par la technique MBS est définie par :

$$x'(t) = \max_{v \in L_t} (x(v)) \quad (3.1)$$

où $L_t = [tI, (t+1)I[$ et I est la taille de l'intervalle d'échantillonnage (exprimée en nombre d'observations à considérer).

3.2.1 Preuve mathématique de la conservation de l'auto-similarité

Soit $Y = \{Y(t); t \geq 0\}$ le processus continu qui représente le volume cumulatif des données qui traversent le lien (exprimé en Mb). On suppose qu'il est auto-similaire d'ordre H .

Par conséquent, le processus discret $X = \{X(t) = (Y(t) - Y(t - 1)); t \in \mathbb{N}\}$, qui représente le débit mesuré à l'intervalle $[t, t + 1[$ et qui est supposé stationnaire, est auto-similaire d'ordre H (d'après le théorème 1.1).

Soit le processus $X' = \left\{ X'(t) = \max_{v \in L_t} (X(v)); t \in \mathbb{N} \right\}$ tel que $L_t = [tI, (t + 1)I[$ et I est la taille de l'intervalle d'échantillonnage.

Objectif : l'objectif est de démontrer que X' est un processus discret auto-similaire. La démonstration consiste à trouver un processus continu $Y' = \{Y'(t); t \geq 0\}$ auto-similaire tel que :

$$X'(t) = Y'(t) - Y'(t - 1) \quad \forall t \in \mathbb{N}. \quad (3.2)$$

Ensuite, grâce au théorème 1.1, on peut déduire que X' est auto-similaire.

La démonstration se compose de trois étapes :

– Première étape : trouver un processus continu $Y'(t)$ tel que

$$Y'(t) - Y'(t - 1) = X'(t) \quad \forall t \in \mathbb{N}.$$

– Deuxième étape : démontrer que $Y'(t)$ est auto-similaire d'ordre H .

– Troisième étape : déduire que $X' = \left\{ X'(t) = \max_{v \in L_t} (X(v)); t \in \mathbb{N} \right\}$ est auto-similaire d'ordre H .

Première étape

On a :

$$\begin{aligned} X'(t) &= \max_{v \in L_t} (X(v)) \text{ et } X'(t) = Y'(t) - Y'(t - 1) \\ &\Rightarrow Y'(t) - Y'(t - 1) = \max_{v \in L_t} (X(v)). \end{aligned}$$

Ainsi, on a :

$$\begin{aligned}
 Y'(t) - Y'(t-1) &= \max_{v \in L_t} (X(v)), \\
 Y'(t-1) - Y'(t-2) &= \max_{v \in L_{t-1}} (X(v)), \\
 &\vdots \\
 Y'(1) - Y'(0) &= \max_{v \in L_1} (X(v)).
 \end{aligned}$$

Si on calcule la somme de ces équations terme à terme, on obtient :

$$Y'(t) - Y'(0) = \sum_{s=1}^t \max_{v \in L_s} (X(v)).$$

Ainsi, on pose $Y'(t)$ tel que :

$$Y'(t) = Y(0) + \sum_{s=1}^t \max_{v \in L_s} (Y(v) - Y(v-1)). \quad (3.3)$$

Deuxième étape

Pour démontrer que $Y'(t)$ est auto-similaire, il suffit de démontrer les théorèmes 3.1, 3.2 et 3.3.

Théorème 3.1. *La somme de deux processus continus $U = \{U(t); t \geq 0\}$ et $W = \{W(t); t \geq 0\}$ auto-similaires d'ordre H est un processus auto-similaire d'ordre H .*

Preuve du théorème 3.1

Puisque U et W sont auto-similaires alors on a :

$$\begin{aligned}
 P[(U + W)(t) \leq z] &= P[U(t) + W(t) \leq z] \\
 &= P[U(t) \leq z - W(t)] \\
 &= P[a^{-H}U(at) \leq z - W(t)] \quad \forall a > 0, \text{ et puisque } U \text{ est auto-similaire} \\
 &= P[W(t) \leq z - a^{-H}U(at)] \quad \forall a > 0 \\
 &= P[a^{-H}W(at) \leq z - a^{-H}U(at)], \quad \text{et puisque } W \text{ est auto-similaire} \\
 &= P[a^{-H}(U + W)(at) \leq z] \quad \forall a > 0.
 \end{aligned}$$

D'où le processus $U + W$ est auto-similaire.

Théorème 3.2. *Soit U et W deux processus continus égaux en distribution, alors on a :*

$$P[\max_{v \in L_t}(U(v)) \leq z] = P[\max_{v \in L_t}(W(v)) \leq z] \quad \forall t > 0.$$

Preuve du théorème 3.2

on a :

$$\max_{v \in L_t}(U(v)) \leq z \quad \Leftrightarrow \quad \forall v \in L_t, \quad U(v) \leq z.$$

Par suite, on a :

$$P[\max_{v \in L_t}(U(v)) \leq z] = P[\forall v \in L_t, U(v) \leq z]. \quad (3.4)$$

De même, il est facile de montrer que :

$$P[\max_{v \in L_t}(W(v)) \leq z] = P[\forall v \in L_t, W(v) \leq z], \quad (3.5)$$

or, puisque U et W sont égaux en distribution, on a :

$$P[\forall v \in L_t, U(v) \leq z] = P[\forall v \in L_t, W(v) \leq z] \quad (3.6)$$

D'après les équations 3.4, 3.5 et 3.6, on peut conclure que :

$$P[\max_{v \in L_t}(U(v)) \leq z] = P[\max_{v \in L_t}(W(v)) \leq z] \quad \forall t > 0. \quad (3.7)$$

Théorème 3.3. *Soit $U = \{U(t); t \geq 0\}$ est un processus continu auto-similaire d'ordre $H \in [\frac{1}{2}, 1[$. Le processus défini par $W = \{W(t) = \max_{v \in L_t}(U(v)); t > 0\}$ est aussi auto-similaire d'ordre H .*

Preuve du théorème 3.3

U est un processus auto-similaire signifie que les variables aléatoires $a^{-H}U(at)$ et $U(t)$ sont égales en distribution (chap. 1, déf. 1.5). En appliquant le théorème 3.2, on a :

$$\begin{aligned} P[\max_{v \in L_t}(a^{-H}U(av)) \leq z] &= P[\max_{v \in L_t}(U(v)) \leq z] \quad \forall a > 0 \\ \Rightarrow P[a^{-H} \max_{v \in L_t}(U(av)) \leq z] &= P[\max_{v \in L_t}(U(v)) \leq z] \\ \Rightarrow P[a^{-H}W(at) \leq z] &= P[W(t) \leq z] \quad \forall a > 0. \end{aligned} \quad (3.8)$$

Par conséquent, W est un processus continu auto-similaire d'ordre H .

Troisième étape

Puisque $Y(t)$ est auto-similaire d'ordre H et d'après les théorèmes 3.1 et 3.3, le processus $Y'(t)$ défini par :

$$Y'(t) = Y(0) + \sum_{s=1}^t \max_{v \in L_s}(Y(v) - Y(v-1)) \quad (3.9)$$

est auto-similaire d'ordre H .

$Y'(t)$ est auto-similaire d'ordre H alors, en utilisant le théorème 1.1, le processus discret X' défini par $X'(t) = Y'(t) - Y'(t-1) = \max_{v \in L_t}(X(v))$ est un processus auto-similaire d'ordre H . Cela prouve que la série échantillonnée par la technique MBS garde l'auto-similarité avec le même ordre de la série originale.

3.3 Résultats expérimentaux

Dans une première partie, nous avons effectué des expériences pour comparer la technique proposée par rapport aux autres techniques d'échantillonnage. Nous montrons que la technique proposée préserve l'auto-similarité et la dépendance à long terme du trafic. Ensuite, nous avons comparé la performance de la prévision obtenue avec des données échantillonnées avec les différentes techniques afin de mettre en évidence l'effet de ces techniques d'échantillonnage sur la performance de la prévision.

3.3.1 Présentation des traces et prétraitement

Pour effectuer ces expérimentations, nous avons utilisé les traces du lien Auckland (sect. 2.2). Trente minutes de traces de données ont été utilisées : quinze minutes pour l'apprentissage et quinze minutes pour l'évaluation de la performance du modèle de prévision. Nous avons extrait le débit de données chaque 10 ms. Les échantillonnages systématique, stratifié et MBS sont appliqués en utilisant différentes granularités (allant de 100 ms à 1000 ms). Nous avons aussi appliqué un lissage pour les données en utilisant la formule (équ. 1.11) pour différentes granularités afin de comparer les données lissées avec le trafic échantillonné.

3.3.2 Comparaison de la technique MBS avec les techniques existantes

Nous avons effectué une série de comparaisons entre les techniques d'échantillonnage par rapport à différentes variables statistiques. Plusieurs expériences sont effectuées en utilisant différentes tailles d'intervalles d'échantillonnage (de 100 ms à 1 s).

La figure 3.2 montre les moyennes des données échantillonnées avec les différentes techniques comparées à la moyenne originale des données. L'échantillonnage stratifié sous-estime la moyenne dans la majorité des cas tandis que l'échantillonnage systématique la surestime. Le lissage du trafic préserve la même moyenne que le trafic (sur la figure, les moyennes des données lissées sont confondues avec celles du trafic original).

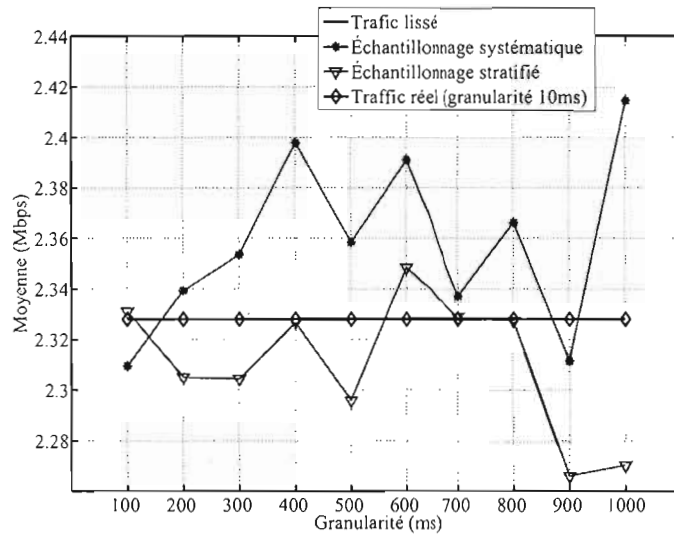


Figure 3.2 Moyenne des données échantillonnées.

Cependant, les moyennes des données échantillonnées avec la technique MBS ne sont pas présentées dans la figure puisque les moyennes des pics du trafic sont largement supérieures à celles des données originales.

La figure 3.3 montre la variance du trafic échantillonné comparée à celle du trafic réel. Elle montre que le lissage du trafic réduit clairement la variance des données. Ceci est inadéquat pour la prévision des pics du trafic et des rafales parce que le modèle de prévision ne pourra pas prendre en considération la variance des données qui sera perdue lors du lissage. Par rapport aux données échantillonnées avec la technique MBS (fig. 3.4), les données lissées ont perdu leur variance. La figure 3.3 montre aussi que la variance des données est préservée pour l'échantillonnage systématique et l'échantillonnage stratifié.

La figure 3.5 montre la sporadicité (PM) par rapport à la granularité pour les données échantillonnées et les données lissées. Elle montre que la sporadicité est élevée pour les données obtenues avec l'échantillonnage systématique et stratifié. Intuitivement, une faible sporadicité rend les données plus prévisibles (c.-à-d. le modèle de prévision

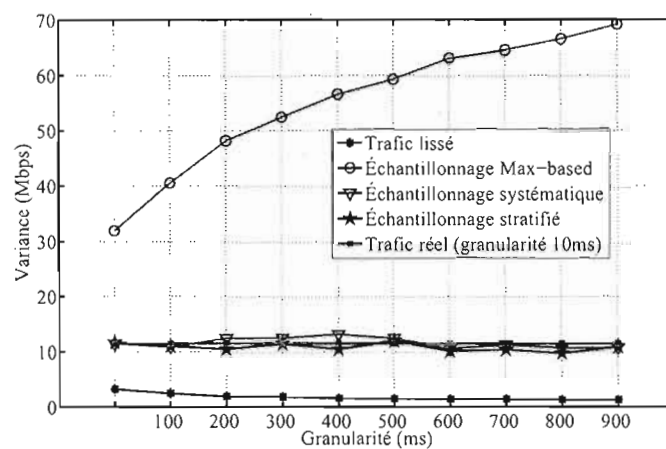


Figure 3.3 Variance des données échantillonnées.

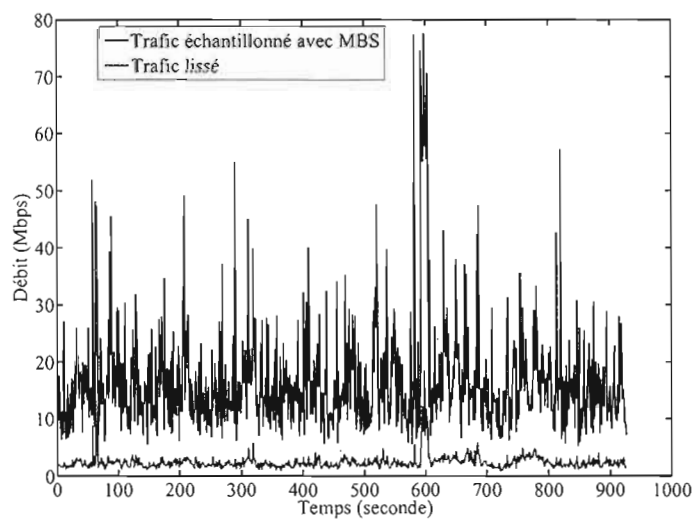


Figure 3.4 Comparaison entre le trafic lissé et le trafic échantillonné avec MBS.

donne une meilleure performance en termes d'erreur). Par conséquent, les données lissées et les données échantillonnées avec la technique MBS devraient améliorer la performance de la prévision (fig. 3.5). Cela va être validé par des prévisions effectuées ultérieurement.

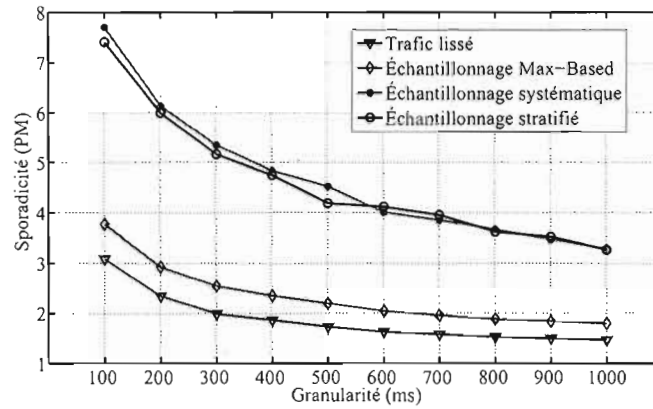


Figure 3.5 Mesure de la sporadicité (PM).

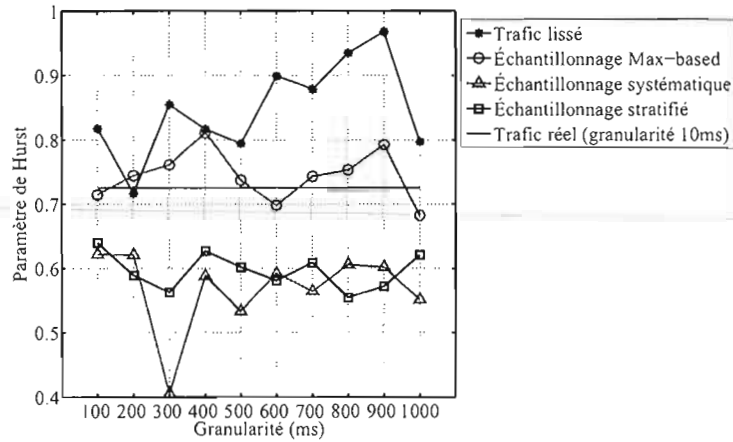


Figure 3.6 Paramètre de Hurst pour différentes granularités.

D'autre part, le paramètre de *Hurst* (fig. 3.6) diffère pour différentes granularités et pour différentes techniques d'échantillonnage. Nous avons utilisé la méthode de décomposition par ondelettes proposée par (Abry et Veitch, 1998) afin d'estimer le paramètre de *Hurst* avec un intervalle de confiance de 95 %. L'idée est de comparer les différentes techniques par rapport à la préservation du paramètre de *Hurst* qui représente

l'auto-similarité du trafic réel. Ainsi, idéalement, la technique d'échantillonnage va conserver la valeur du paramètre de *Hurst* du trafic réel ($H \approx 0,72$). La figure montre que le paramètre de *Hurst* a baissé lors de l'échantillonnage systématique et stratifié par rapport au trafic réel. Il décroît en dessous de 0,6 (granularité 300 ms) ce qui signifie que les données échantillonnées ont perdu de leur auto-similarité. Cela est dû au fait que les techniques d'échantillonnage systématique et stratifié sont basées sur un échantillonnage aléatoire des données.

Les données échantillonnées avec la technique MBS ont conservé leur auto-similarité pour toutes les granularités (par rapport au trafic réel). Cela confirme les résultats mathématiques de la préservation de l'auto-similarité pour la technique MBS.

D'autre part, le lissage du trafic a augmenté le paramètre de *Hurst*, c.-à-d. l'auto-similarité et la dépendance à long terme. Ainsi, nous pouvons s'attendre à ce que le trafic sera plus prévisible pour des données échantillonnées avec la technique MBS et pour des données lissées.

On note aussi que les résultats obtenus montrent que le paramètre de *Hurst* (qui reflète le degré d'auto-similarité) ne traduit pas la sporadicité du trafic. En effet, les techniques d'échantillonnage systématique et stratifié ont augmenté la sporadicité des données (fig. 3.5), mais ils ont réduit son auto-similarité (fig. 3.6). D'autre part, le lissage du trafic ou l'échantillonnage MBS ont réduit la sporadicité, mais ils ont augmenté l'auto-similarité. Cette observation est importante pour les futures analyses du trafic afin de tenir compte de la sporadicité et l'auto-similarité comme deux paramètres différents.

3.3.3 Exploitation des techniques d'échantillonnage pour la prévision du trafic

Nous effectuons un ensemble d'expérimentations en utilisant les données échantillonnées à partir des données du lien Auckland (sect. 2.2.1). Nous appliquons les trois techniques d'échantillonnage ainsi que le lissage du trafic sur les données avant de les utiliser pour la prévision. L'objectif est de comparer la performance de la prévision pour étudier l'effet de la technique d'échantillonnage.

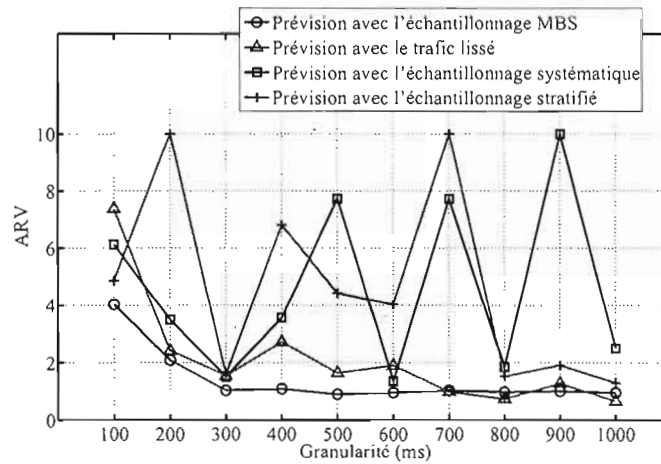


Figure 3.7 Prédiction avec le modèle LMMSE en utilisant les données échantillonnées.

La figure 3.7 montre l'erreur de prédiction (ARV) obtenue en utilisant le modèle de prédiction LMMSE pour différentes granularités. Chaque courbe représente l'ARV obtenue en utilisant des données échantillonnées avec une des différentes techniques étudiées. On remarque que la prédiction du trafic, en utilisant le trafic lissé, offre une meilleure performance comparée à l'échantillonnage stratifié ou systématique. La performance obtenue avec ces deux techniques d'échantillonnage est très variable d'une granularité à une autre. Pour l'échantillonnage MBS, la figure montre que l'erreur de la prédiction est toujours inférieure à celles des autres techniques pour toutes les granularités. Ceci confirme les observations effectuées dans la section précédente. En effet, les techniques d'échantillonnage systématique et stratifié ont fait perdre au trafic son auto-similarité et sa dépendance à long terme en gardant une importante sporadicité. Par conséquent, les données sont devenues moins prévisibles. Par contre, le trafic lissé et le trafic échantillonné par la technique MBS ont conservé l'auto-similarité tout en réduisant la sporadicité. Cela peut expliquer l'amélioration de la performance des prévisions en utilisant le trafic lissé ou échantillonné avec la technique MBS.

Le tableau 3.1 présente un résumé des différents résultats obtenus. Il compare les techniques d'échantillonnage par rapport à la préservation des paramètres statistiques,

Tableau 3.1 Résumé des résultats obtenus

-	Moyenne	Variance	Auto-similarité	Sporadicité	Les pics du trafic	Performance de la prévision
Échantillonnage MBS	Traite les pics	Importante	Préservée	Réduite	Préservés	Bonne
Échantillonnage systématique	Perdue	Préservée	Perdue	Importante	Perdus	Très variable
Échantillonnage stratifié	Perdue	Préservée	Perdue	Importante	Perdus	Très variable
Lissage	Préservée	Réduite	Augmentée	Réduite	Perdus	Bonne

l'auto-similarité, la sporadicité et leur effet sur la performance de la prévision. On note que l'échantillonnage MBS ne conserve ni la moyenne ni la variance, mais il préserve l'auto-similarité du trafic. Il améliore aussi la performance de la prévision pour prévoir les pics du trafic. Le lissage du trafic augmente l'auto-similarité, mais il offre une performance de prévision acceptable. Les autres techniques d'échantillonnage ne préservent pas la moyenne, mais plutôt la variance. Leur utilisation donne des performances de prévision très variables. D'autre part, l'échantillonnage MBS garantit que les données échantillonnées gardent les pics du trafic afin de les prévoir. Cela permet de réserver les ressources futures ou gérer les files d'attente avec plus de précision en tenant compte des hausses du trafic.

3.4 Conclusion

Dans ce chapitre, nous avons étudié l'utilisation des techniques d'échantillonnage pour le débit. Nous avons montré que l'application des techniques d'échantillonnage a un effet considérable sur les caractéristiques des données obtenues et sur la performance

du modèle de prévision.

Nous avons aussi proposé une nouvelle technique d'échantillonnage qui peut être utilisée avec les modèles de prévision afin d'améliorer leurs performances. Nous avons aussi montré que la technique d'échantillonnage MBS préserve l'auto-similarité, la dépendance à long terme et les pics du trafic. Ainsi, les modèles de prévision, combinés avec la technique d'échantillonnage proposée, peuvent prévoir avec précision les pics de trafic afin de les utiliser pour la réservation des ressources et pour la gestion des files d'attente.

Dans les chapitres précédents, nous avons étudié les différents paramètres et techniques qui peuvent améliorer la performance de la prévision en termes d'erreur. Dans ce qui suit, nous proposons des applications pour la prévision dans le cadre de la gestion des files d'attente et le contrôle du taux de perte dans les réseaux optiques à commutation de rafales.

CHAPITRE IV

PROPOSITION D'UN MÉCANISME DE GESTION DE FILES D'ATTENTE BASÉE SUR LA PRÉVISION

«*Agir en primitif et prévoir en stratège*»

René Char

La gestion des files d'attente traditionnelles dans les réseaux se base sur la discipline *premier arrivé premier servi* (*First In First Out* - FIFO). Celle-ci a beaucoup de limitations telles que la saturation continue des files d'attente, l'augmentation du taux de perte, la synchronisation globale des différents flux de données qui parcourent le réseau, la sous-utilisation du réseau et la non équité entre les différents flux (Ren, Lin et Liu, 2003).

Afin de résoudre ces problèmes, des mécanismes de gestion de files d'attente (*Active Queue Management* - AQM) ont été proposés (RED, PAQM, ASPACE, etc.). Ils permettent d'éviter la congestion des files d'attente et se basent sur le rejet proactif (à l'avance) des paquets pour prévenir les sources de congestion du réseau. Cependant, ces mécanismes présentent des défauts majeurs. En effet, ils dépendent des paramètres définis par l'administrateur du réseau, ce qui les rend moins flexibles aux variations du trafic. Par conséquent, leurs avantages ne sont pas toujours garantis, notamment en termes de stabilité de la taille de la file, de taux de perte et de la quantité de données reçues avec succès par la destination.

Ainsi, nous proposons un nouveau mécanisme de gestion de files d'attente, bap-

tisé α -SNFAQM, qui est basé sur la prévision neurofloue du trafic (Zhani, Elbiaze et Kamoun, 2007; Zhani, Elbiaze et Kamoun, 2009a). Ce mécanisme permet de stabiliser la taille de la file d'attente tout en garantissant une bonne utilisation du lien et en maximisant les données reçues avec succès par la destination. Il est évalué par une étude analytique pour calculer la taille moyenne de la file d'attente. Des simulations ont été aussi effectuées afin de le comparer à d'autres mécanismes (RED, APACE, PAQM). Dans ce qui suit, nous commençons par présenter le mécanisme proposé.

4.1 α -SNFAQM

Avant de présenter le principe de fonctionnement du mécanisme proposé, nous définissons quelques paramètres :

- *L'intervalle de contrôle* est l'intervalle de temps qui sépare deux mises-à-jour successives des variables de l'algorithme. On note par I la taille de cet intervalle. Ainsi, les variables de l'algorithme sont mises à jour chaque I secondes. La valeur de I est idéalement choisi comme un multiple du temps d'aller retour (RTT) moyen dans le réseau. En effet, les sources TCP réagissent après chaque RTT (détection des pertes, retransmission, etc.). Dans ce qui suit, l'intervalle τ signifie l'intervalle de temps entre les instants $\tau \times I$ et $(\tau + 1) \times I$, c.-à-d. l'intervalle $[\tau \times I, (\tau + 1) \times I[$.
- Le débit d'entrée des données (*Input Rate*), noté $IR(\tau - 1)$, est la quantité de données (mesurée en octets) qui arrive à la file d'attente à l'intervalle $(\tau - 1)$.
- Le débit d'entrée prédit (*Predicted Input Rate*), noté $PIR(\tau)$, est la quantité de données (mesurée en octets) qui est prédite pour l'intervalle τ . $PIR(\tau + 1)$ est la prévision du débit d'entrée pour l'intervalle $(\tau + 1)$.
- Le débit de sortie (*Output Rate*), noté OR , est la quantité maximale de données

(mesurée en octets) qui pourrait être servie par la file d'attente pendant l'intervalle de contrôle. Il représente la capacité de service de la file pendant cet intervalle. La valeur du débit de sortie OR est fixe et ne dépend que de la capacité du lien de sortie et de la taille de l'intervalle de contrôle. Ainsi, on a

$$OR = C \times I \quad (4.1)$$

où C est la capacité du lien de sortie.

4.1.1.1 Principe de fonctionnement

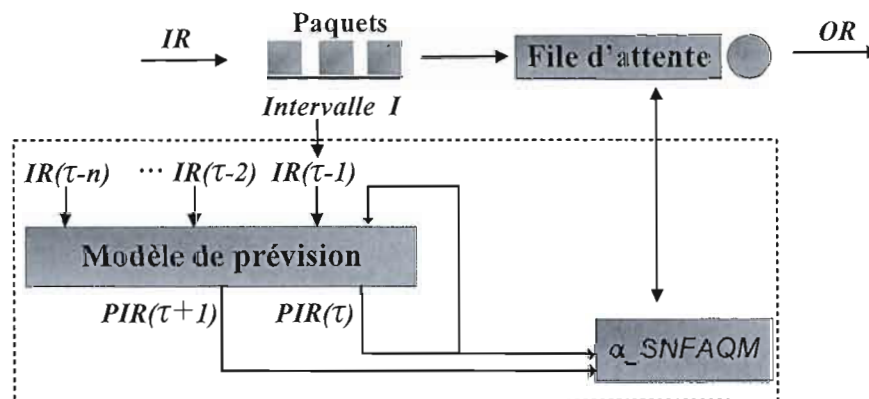


Figure 4.1 Principe de fonctionnement du mécanisme α -SNFAQM.

Le débit d'entrée IR est mesuré pour chaque intervalle de temps de taille I . Le modèle de prévision calcule le débit d'entrée des données dans le futur (fig. 4.1). Ainsi, il calcule $PIR(\tau)$ en utilisant les n derniers retards (c.-à-d. $IR(\tau-i)$ où $1 \leq i \leq n$) comme variables d'entrée du modèle de prévision. Il calcule aussi $PIR(\tau+1)$ en utilisant les retards $IR(\tau-i)$ où $1 \leq i \leq n-1$ et $PIR(\tau)$ comme entrées. Le modèle de prévision utilisé est le modèle α -SNF. Cependant, il peut être remplacé par un autre modèle de prévision.

Ensuite, $PIR(\tau)$ et $PIR(\tau+1)$ sont utilisés par le mécanisme α -SNFAQM pour

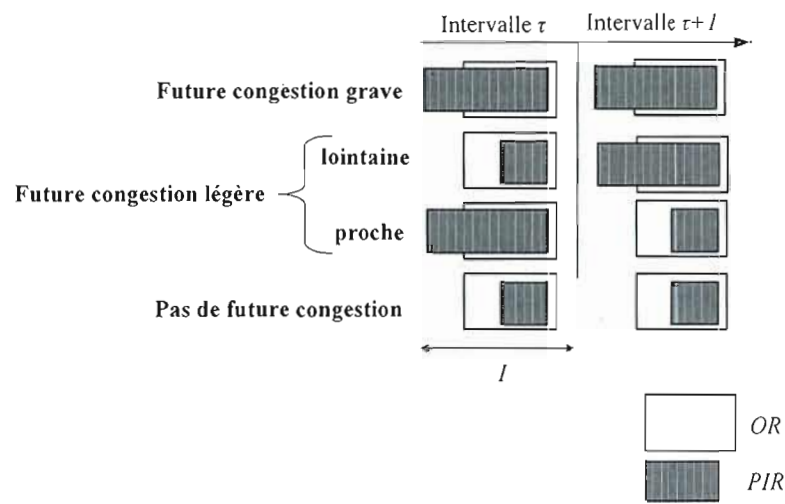


Figure 4.2 Congestion grave et congestion légère.

prévoir les futures congestions et distinguer deux niveaux de congestion : congestion « grave » et congestion « légère » (fig. 4.2).

La figure 4.2 illustre comment l'algorithme α -SNFAQM prévoit la congestion. Le débit d'entrée est prédit pour deux intervalles consécutifs $PIR(\tau)$ et $PIR(\tau+1)$. Le débit d'entrée PIR est ensuite comparé à la quantité maximale de données qui peut être servie par la file, c.-à-d. OR . Si le débit d'entrée prédit dépasse la capacité de service de la file pour deux intervalles consécutifs alors il y aura une future congestion grave. Si le débit d'entrée prédit dépasse la capacité de service de la file pour un des deux prochains intervalles de temps alors il y aura une future congestion légère qui peut être proche ou lointaine (fig. 4.2).

4.1.2 Algorithme

La figure 4.3 montre l'algorithme α -SNFAQM et les différentes décisions de rejet ou d'acceptation des paquets.

La variable *avg* représente la taille moyenne de la file d'attente exprimée en

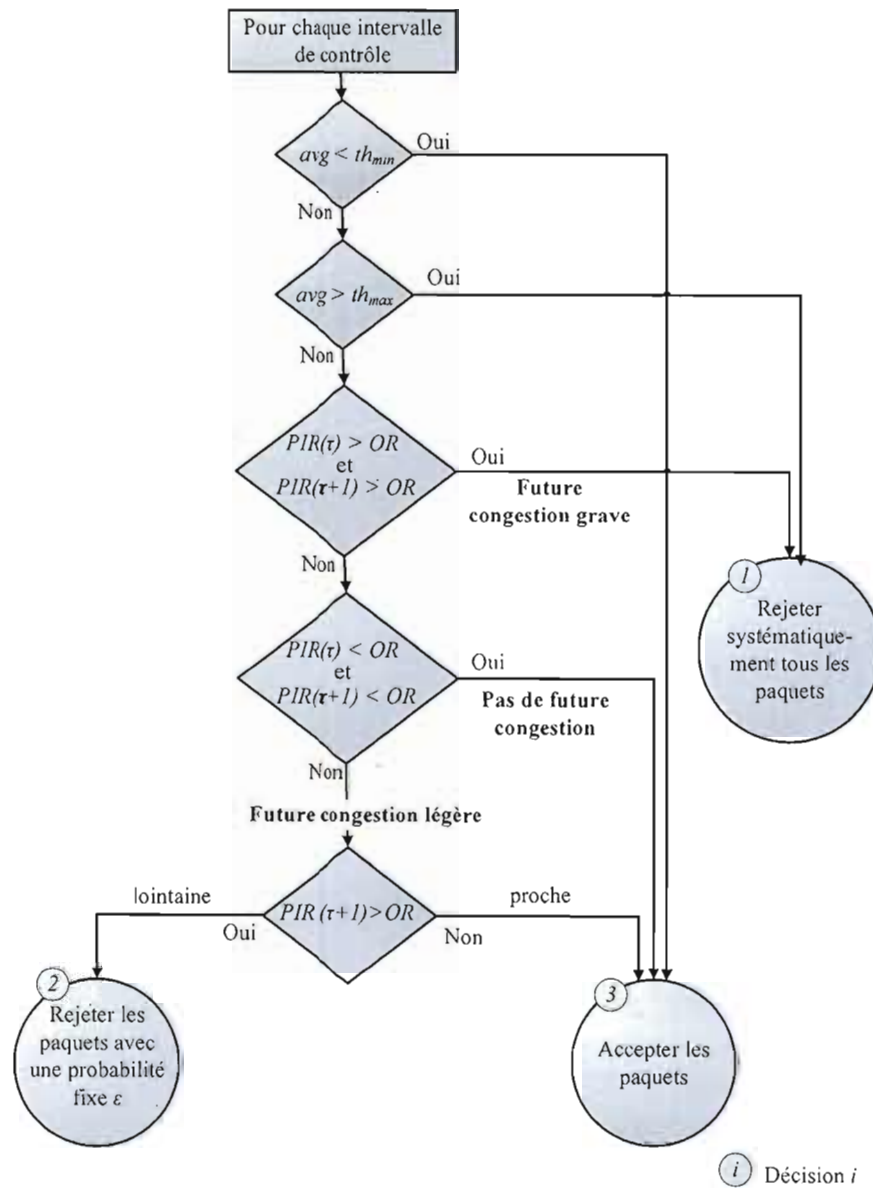


Figure 4.3 Algorithme α -SNFAQM.

nombre de paquets telle que :

$$avg = (1 - w_q)avg + w_q q, \quad (4.2)$$

où $w_q \in]0, 1[$ est un paramètre fixé par l'administrateur et q est la taille instantanée de la file d'attente exprimée en nombre de paquets. La valeur de la taille moyenne avg est mise à jour à l'arrivée de chaque paquet. Cette formule est utilisée aussi par le mécanisme RED pour estimer la taille moyenne de la file (app. B). Soit th_{min} le seuil minimal de la taille de la file d'attente et th_{max} son seuil maximal exprimés en nombre de paquets. Si la taille moyenne de la file est inférieure à th_{min} , tous les paquets sont acceptés. Si elle est supérieure à th_{max} , tous les paquets sont rejetés. Si elle est entre les seuils th_{min} et th_{max} , le mécanisme α -SNFAQM a recours à la prévision pour décider si les paquets doivent être rejetés ou acceptés.

Selon l'état futur de la file d'attente, le mécanisme α -SNFAQM prend la décision adéquate. Dans ce qui suit, nous présentons les différents états futurs possibles de la file d'attente ainsi que la décision prise pour chaque cas.

- Future congestion grave

On considère qu'il y aura une future congestion grave si la file d'attente sera congestionnée pendant les deux prochains intervalles (fig. 4.2). Par conséquent, durant les deux prochains intervalles τ et $(\tau+1)$, la file d'attente ne sera pas capable de servir les données qui arriveront à la file. Dans ce cas, la décision est de rejeter systématiquement tous les paquets qui arrivent à la file (fig. 4.3 , Décision 1). Ainsi, une future congestion grave se produira quand :

$$PIR(\tau) > OR \text{ et } PIR(\tau + 1) > OR. \quad (4.3)$$

- Future congestion légère

On considère qu'il y aura une future congestion légère si la file sera congestionnée pour l'un des deux prochains intervalles τ et $(\tau + 1)$ (fig. 4.2). Il est sûr que des paquets seront rejetés à l'intervalle τ ou $(\tau + 1)$. Ainsi, il y a deux cas :

- Une future congestion légère proche : la file sera congestionnée seulement à l'intervalle τ (fig. 4.2). Dans ce cas, la décision est de ne pas rejeter les paquets (fig. 4.3, Décision 3) puisqu'ils pourront être servis à l'intervalle suivant. Ainsi, il y aura une future congestion légère proche si

$$PIR(\tau) > OR \text{ et } PIR(\tau + 1) < OR. \quad (4.4)$$

- Une future congestion légère lointaine : la file sera congestionnée seulement à l'intervalle $(\tau + 1)$, c'est-à-dire $PIR(\tau + 1) > OR$ (fig. 4.2). Dans ce cas, la décision est de rejeter les paquets avec une probabilité fixe, notée ϵ , dont la valeur est fixée par l'administrateur ; cela permet d'éviter la congestion dans le prochain intervalle (fig. 4.3 , Décision 2). Ainsi, il y aura une future congestion légère lointaine si

$$PIR(\tau) < OR \text{ et } PIR(\tau + 1) > OR. \quad (4.5)$$

- Pas de future congestion

Il n'y aura pas de future congestion si la file d'attente ne sera pas congestionnée pendant les deux prochains intervalles τ et $(\tau + 1)$ (fig. 4.2). Par conséquent, la file d'attente sera capable de servir les paquets qui seront reçus. Ainsi, la décision est d'accepter tous les paquets qui arrivent à la file (fig. 4.3 , Décision 3). Il n'y aura pas de congestion future si :

$$PIR(\tau) < OR \text{ et } PIR(\tau + 1) < OR. \quad (4.6)$$

4.2 Étude analytique

On se propose de calculer analytiquement les probabilités de réalisation des différents types de congestion (sans l'effet du mécanisme α -SNFAQM) au niveau d'une

file d'attente. Cela est réalisé en utilisant une modélisation du trafic au niveau de la couche TCP (généralement, plus de 80 % du trafic est constitué par des flux TCP (chap. 2)). Nous analysons ensuite le fonctionnement du α -SNFAQM en se basant sur l'étude analytique. Ensuite, nous calculons analytiquement la probabilité de réalisation pour chaque état futur et la taille moyenne de la file d'attente avec et sans le mécanisme α -SNFAQM.

Le tableau 4.1 montre la notation pour chaque état de congestion.

Tableau 4.1 Notation pour chaque état

État	Notation
Future congestion grave	1
Future congestion légère proche	2
Future congestion légère lointaine	3
Pas de future congestion	4

On définit la probabilité de réalisation de chaque état de congestion i , notée P_i , par :

$$P_1 = P(PIR(\tau) > OR \text{ et } PIR(\tau + 1) > OR), \quad (4.7)$$

$$P_2 = P(PIR(\tau) > OR \text{ et } PIR(\tau + 1) < OR), \quad (4.8)$$

$$P_3 = P(PIR(\tau) < OR \text{ et } PIR(\tau + 1) > OR), \quad (4.9)$$

$$P_4 = P(PIR(\tau) < OR \text{ et } PIR(\tau + 1) < OR), \quad (4.10)$$

$$\sum_{i=1}^4 P_i = 1. \quad (4.11)$$

Nous présentons d'abord les hypothèses de l'analyse avant de calculer les probabilités de réalisation des états futures de congestion.

Hypothèses :

1. La prévision est parfaite c'est-à-dire l'erreur de la prévision est nulle. Par suite, on a

$$PIR(\tau) = IR(\tau) \quad \forall \tau. \quad (4.12)$$

2. La version TCP utilisée est la version Reno 1 (Allman, Paxson et Stevens, 1999). Ainsi, le débit d'une connexion i à l'intervalle $(\tau + 1)$, noté par $x_i(\tau + 1)$, est estimé par (Low, 2003) :

$$x_i(\tau + 1) = x_i(\tau) + \frac{1 - q_i(\tau)}{D_i^2} - \frac{2}{3} q_i(\tau) x_i^2(\tau), \quad (4.13)$$

où $q_i(\tau)$ représente le taux de perte pour la connexion i et D_i représente le temps d'aller-retour d'un paquet appartenant à cette connexion.

3. On suppose que toutes les connexions TCP ont le même temps d'aller-retour, noté D , donc on a $D_i = D \quad \forall i$.
4. On suppose que toutes les connexions TCP ont le même taux de perte, donc on a $q_i(\tau) = q(\tau)$.
5. On suppose que $IR(\tau)$ est uniformément distribué dans l'intervalle $[0, IR_{max}]$ où IR_{max} est le débit maximal d'entrée de la file d'attente (app. A, déf. A.4).
6. On a :

$$IR(\tau + 1) = \sum_i^n x_i(\tau + 1) \quad (4.14)$$

où n est le nombre de connexions TCP existantes.

En utilisant l'équation 4.13, on a :

$$IR(\tau + 1) = \sum_i^n \left(x_i(\tau) + \frac{1 - q_i(\tau)}{D_i^2} - \frac{2}{3 D_i} q_i(\tau) x_i^2(\tau) \right),$$

donc

$$IR(\tau + 1) = \sum_i^n x_i(\tau) + \sum_i^n \frac{1 - q_i(\tau)}{D_i^2} - \frac{2}{3} \sum_i^n q_i(\tau) x_i^2(\tau).$$

Par suite, on a :

$$IR(\tau + 1) = IR(\tau) + \frac{1}{D^2} \sum_i^n (1 - q_i(\tau)) - \frac{2}{3} \sum_i^n q_i(\tau) x_i^2(\tau). \quad (4.15)$$

Puisque $q_i(\tau) = q(\tau)$ et $D_i = D$, l'équation 4.15 devient :

$$IR(\tau + 1) = IR(\tau) + \frac{n(1 - q(\tau))}{D^2} - \frac{2}{3}q(\tau) \sum_{i=1}^n x_i^2(\tau). \quad (4.16)$$

D'autre part, on a :

$$\sum_i^n x_i^2(\tau) = \left(\sum_{i=1}^n x_i(\tau)\right)^2 - 2 \sum_{i < j} x_i(\tau)x_j(\tau). \quad (4.17)$$

En supposant que les débits de deux connexions TCP sont quasiment égaux (c.-à-d. $x_i \approx x_j$), on a :

$$\frac{\sum_{i < j} x_i(\tau)x_j(\tau)}{(n/2)(n-1)} \approx \frac{\sum_{i=1}^n x_i^2(\tau)}{n}. \quad (4.18)$$

Ainsi,

$$\sum_{i < j} x_i(\tau)x_j(\tau) \approx \frac{(n-1)}{2} \sum_{i=1}^n x_i^2(\tau). \quad (4.19)$$

Par suite, l'équation 4.17 devient

$$\sum_i^n x_i^2(\tau) = \left(\sum_{i=1}^n x_i(\tau)\right)^2 - 2 \frac{(n-1)}{2} \sum_{i=1}^n x_i^2(\tau) \quad (4.20)$$

$$\Leftrightarrow \sum_i^n x_i^2(\tau) = \frac{1}{n} \left(\sum_{i=1}^n x_i(\tau)\right)^2 = \frac{IR^2(\tau)}{n}. \quad (4.21)$$

En utilisant l'équation 4.16, l'équation 4.17 devient

$$IR(\tau + 1) = IR(\tau) + \frac{n(1 - q(\tau))}{D^2} - \frac{2}{3} \frac{q(\tau)}{n} IR^2(\tau). \quad (4.22)$$

Chaque courbe dans la figure 4.4 montre $IR(\tau + 1)$ en fonction de $IR(\tau)$ pour un taux de perte donné $q(\tau)$ (avec $D = 400$ ms, $n = 20$, $OR = 160$ octets). Cette figure est obtenue en fixant, pour chaque courbe, la valeur de $q(\tau)$ dans l'équation 4.22.

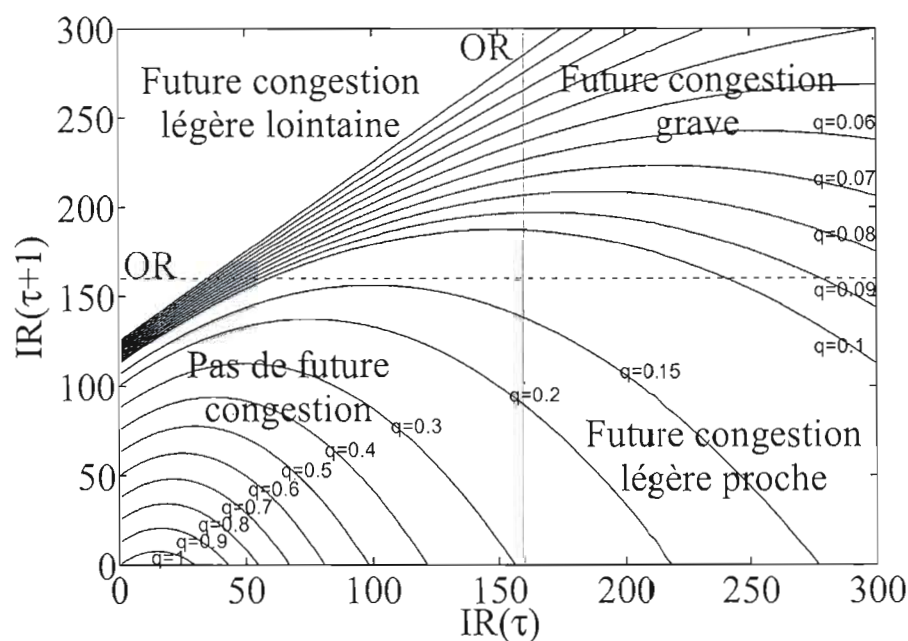


Figure 4.4 État futur de la file par rapport au taux de perte $q(\tau)$ et le débit d'entrée $IR(\tau)$.

En considérant les équations 4.3, 4.4, 4.5 et 4.6, qui représentent les quatre états futurs de la file d'attente, on peut diviser le plan de la figure 4.4 sur quatre surfaces. Chaque surface représente un état futur de la file d'attente.

Ainsi, à l'aide de la figure, connaissant l'état actuel défini par $IR(\tau)$ et $q(\tau)$, on peut déterminer l'état futur de la file d'attente. Par exemple, si le débit d'entrée $IR(\tau)$ est de 250 octets et le taux de perte est de (0,15), alors $IR(\tau + 1)$ est égale à 40 octets. Ainsi, la file subira une congestion légère proche (puisque le point de coordonnées $IR(\tau)$ et $IR(\tau + 1)$ se trouve dans la surface correspondante à cet état).

Ceci montre que l'état futur de la file d'attente (à l'intervalle $\tau + 1$) peut être déterminé seulement en fonction de $q(\tau)$ et $IR(\tau)$.

Dans ce qui suit, nous exprimons les conditions de réalisation de chaque état futur en fonction des variables $IR(\tau)$ et $q(\tau)$ mesurées à l'intervalle τ , c'est-à-dire indépendamment de l'intervalle $\tau + 1$. Ceci est possible grâce à l'équation 4.22.

– **Future congestion grave :**

On a une future congestion grave si

$$IR(\tau) > OR \text{ et } IR(\tau + 1) > OR.$$

En utilisant l'équation 4.22, on a :

$$IR(\tau) > OR \text{ et } IR(\tau) + \frac{n(1 - q(\tau))}{D^2} - \frac{2}{3} \frac{q(\tau)}{n} IR^2(\tau) > OR \quad (4.23)$$

$$\Leftrightarrow IR(\tau) > OR \text{ et } IR(\tau) + \frac{n}{D^2} - \frac{nq(\tau)}{D^2} - \frac{2}{3} \frac{q(\tau)}{n} IR^2(\tau) > OR \quad (4.24)$$

$$\Leftrightarrow IR(\tau) > OR \text{ et } q(\tau) < \frac{IR(\tau) + \frac{n}{D^2} - OR}{\frac{n}{D^2} + \frac{2}{3} \frac{1}{n} IR^2(\tau)}. \quad (4.25)$$

Pour simplifier la notation, on pose :

$$\sigma(IR(\tau)) = \max\left(\frac{IR(\tau) + \frac{n}{D^2} - OR}{\frac{n}{D^2} + \frac{2}{3} \frac{1}{n} IR^2(\tau)}, 0\right). \quad (4.26)$$

Par suite, on a :

$$IR(\tau) > OR \text{ et } q(\tau) < \sigma(IR(\tau)). \quad (4.27)$$

– **Future congestion légère proche :**

On a une future congestion légère proche si

$$IR(\tau) > OR \text{ et } IR(\tau + 1) < OR.$$

En utilisant l'équation 4.22, on a :

$$IR(\tau) > OR \text{ et } IR(\tau) + \frac{n(1-q(\tau))}{D^2} - \frac{2}{3} \frac{q(\tau)}{n} IR^2(\tau) < OR.$$

Cela signifie que :

$$IR(\tau) > OR \text{ et } q(\tau) > \sigma(IR(\tau)). \quad (4.28)$$

– **Future congestion légère lointaine :**

On a une future congestion légère lointaine si

$$IR(\tau) < OR \text{ et } IR(\tau + 1) > OR.$$

En utilisant l'équation 4.22, on a :

$$IR(\tau) < OR \text{ et } IR(\tau) + \frac{n(1-q(\tau))}{D^2} - \frac{2}{3} \frac{q(\tau)}{n} IR^2(\tau) > OR.$$

Cela signifie que :

$$IR(\tau) < OR \text{ et } q(\tau) < \sigma(IR(\tau)). \quad (4.29)$$

– **Pas de future congestion :**

Il n'y a pas de future congestion si :

$$IR(\tau) < OR \text{ et } IR(\tau + 1) < OR.$$

En utilisant l'équation 4.22, on a :

$$IR(\tau) < OR \text{ et } IR(\tau) + \frac{n(1-q(\tau))}{D^2} - \frac{2}{3} \frac{q(\tau)}{n} IR^2(\tau) < OR.$$

Cela signifie que :

$$IR(\tau) < OR \text{ et } q(\tau) > \sigma(IR(\tau)). \quad (4.30)$$

Tableau 4.2 Caractérisation des états futurs de la file d'attente

État futur	Condition par définition	Condition avec la modélisation TCP	Décision du α _SNFAQM	Conséquence
Future congestion grave	$IR(\tau) > OR$ et $IR(\tau + 1) > OR$	$IR(\tau) > OR$ et $q(\tau) < \sigma(IR(\tau))$	Rejet systématique	$IR \searrow$ et $q \nearrow$
Future congestion légère proche	$IR(\tau) > OR$ et $IR(\tau + 1) < OR$	$IR(\tau) > OR$ et $q(\tau) > \sigma(IR(\tau))$	Pas de rejet	$IR \searrow$
Future congestion légère lointaine	$IR(\tau) < OR$ et $IR(\tau + 1) > OR$	$IR(\tau) < OR$ et $q(\tau) < \sigma(IR(\tau))$	Rejet avec probabilité ϵ	$IR \searrow$ et $q \nearrow$
Pas de future congestion	$IR(\tau) < OR$ et $IR(\tau + 1) < OR$	$IR(\tau) < OR$ et $q(\tau) > \sigma(IR(\tau))$	Pas de rejet	$IR \nearrow$

Le tableau 4.2 résume les états futurs de la file et leurs conditions de réalisation en fonction $IR(\tau)$, $\sigma(IR(\tau))$ et OR . Il montre la condition équivalente trouvée par la modélisation TCP pour chaque état.

La figure 4.5 montre $\sigma(IR(\tau))$ en fonction de $IR(\tau)$. En examinant la condition de réalisation de chaque état futur (Tableau 4.2), on peut diviser le plan de la figure 4.5 sur quatre surfaces séparées par la courbe $\sigma(IR(\tau))$ et la ligne rouge (représentant OR). Chaque surface représente un état futur. Ainsi, l'état actuel de la file est défini par le point E de coordonnées $(IR(\tau), q(\tau))$, qu'on peut marquer sur la figure 4.5. L'état futur correspond au nom de la surface à laquelle ce point appartient. Par exemple, si $IR(\tau) = 50$ et $q(\tau) = 0,4$, en examinant le point E de coordonnées $(50, 0,4)$, on trouve qu'il appartient à la surface correspondante à l'état « pas de future congestion ». Ainsi, il n'y aura pas de future congestion.

Le tableau 4.2 montre la décision du mécanisme α _SNFAQM et sa conséquence pour chaque état futur.

Lorsqu'une future congestion grave est prédite, le mécanisme α _SNFAQM aug-

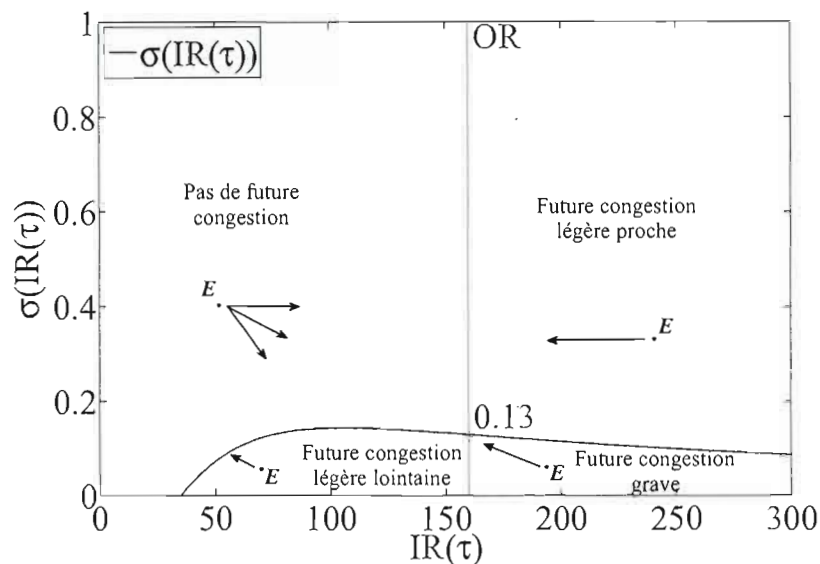


Figure 4.5 La fonction $\sigma(IR(\tau))$ ($n = 20$).

mente le taux de perte q ce qui oblige les sources TCP à réduire le débit d'entrée de la file IR . Par conséquent, le point E représentant l'état actuel de la file se déplacera vers les trois autres surfaces. Ainsi, la prévision suivante donnera probablement l'un des trois autres états futurs (fig. 4.5).

Lorsqu'une congestion légère proche est prédite, le mécanisme α -SNFAQM accepte les paquets puisque le débit d'entrée va être réduit ($IR(\tau + 1) < OR$). Ainsi, le point E , représentant l'état actuel, se déplacera vers les surfaces « future congestion légère lointaine » ou « pas de future congestion ». Par conséquent, la prévision suivante donnera l'un de ces deux états futurs.

Lorsqu'une congestion légère lointaine est prédite, le mécanisme α -SNFAQM rejette les paquets avec une probabilité ϵ ; cela augmente le taux de perte q et oblige les sources TCP à réduire le débit d'entrée de la file. Ainsi, le point E , représentant l'état actuel, se déplacera vers la gauche. Par conséquent, la prévision suivante sera « une congestion légère lointaine » ou « pas de future congestion ».

Lorsque le mécanisme prévoit qu'il n'y aura pas de congestion future, le mécanisme α -SNFAQM accepte les paquets. Ainsi, puisque $(IR(\tau+1) < OR)$, la prévision suivante sera « une congestion légère lointaine » ou « pas de future congestion ». Le point E se déplacera vers ces deux états.

– Probabilité de réalisation des états

Dans ce qui suit, on se propose de calculer les probabilités de réalisation de chaque état. Afin de simplifier les notations, nous omettons le symbole τ puisque les probabilités sont calculées pour un même intervalle τ . Ainsi, les variables $\sigma(IR(\tau))$ et $q(\tau)$ sont notées $\sigma(IR)$ et q .

Le débit maximal d'entrée de la file d'attente IR_{max} dépend de la capacité du lien d'entrée à la file. On note aussi IR_0 et IR_1 les valeurs de IR telles que $\sigma(IR_0) = 0$, $\sigma(IR_1) = 1$ et $IR_0 < IR_1$.

Puisque $IR(\tau)$ est uniformément distribué, la probabilité d'avoir un état futur est estimée par la division de l'aire de la surface correspondante à l'état en question par l'aire de la surface totale (l'union des quatre surfaces) (fig. 4.5).

Ainsi, les probabilités des différents états sont données par :

$$P_1 = \frac{\int_{OR}^{IR_{max}} \sigma(IR).dIR}{\int_0^{IR_{max}} 1.dIR}, \quad (4.31)$$

$$P_2 = \frac{\int_{OR}^{IR_{max}} (1 - \sigma(IR)).dIR}{\int_0^{IR_{max}} 1.dIR}, \quad (4.32)$$

$$P_3 = \frac{\int_0^{OR} \sigma(IR).dIR}{\int_0^{IR_{max}} 1.dIR}, \quad (4.33)$$

$$P_4 = \frac{\int_0^{OR} (1 - \sigma(IR)).dIR}{\int_0^{IR_{max}} 1.dIR}. \quad (4.34)$$

Le développement de ces équations donne :

$$P_1 = \frac{\frac{1}{2C} \left[\log(x^2 + \frac{B}{C}) \right]_{a_2}^{b_2} + \frac{A}{\sqrt{BC}} [\argtan(x)] \sqrt{\frac{C}{B}}^{b_2}}{IR_{max}}, \quad (4.35)$$

$$P_2 = \frac{IR_{max} - OR - \frac{1}{2C} \left[\log(x^2 + \frac{B}{C}) \right]_{a_2}^{b_2} - \frac{A}{\sqrt{BC}} [\argtan(x)] \sqrt{\frac{C}{B}}^{b_2}}{IR_{max}}, \quad (4.36)$$

$$P_3 = \frac{\frac{1}{2C} \left[\log(x^2 + \frac{B}{C}) \right]_{a_1}^{b_1} + \frac{A}{\sqrt{BC}} [\argtan(x)] \sqrt{\frac{C}{B}}^{b_1}}{IR_{max}}, \quad (4.37)$$

$$P_4 = \frac{OR - \frac{1}{2C} \left[\log(x^2 + \frac{B}{C}) \right]_{a_1}^{b_1} - \frac{A}{\sqrt{BC}} [\argtan(x)] \sqrt{\frac{C}{B}}^{b_1}}{IR_{max}}, \quad (4.38)$$

avec

$$A = \frac{n}{D^2} - OR, \quad B = \frac{n}{D^2}, \quad C = \frac{2}{3n},$$

$$[f(x)]_a^b = \begin{cases} f(b) - f(a) & [a, b] \neq \emptyset \\ 0 & [a, b] = \emptyset. \end{cases}$$

Les valeurs de a_i , b_i sont telles que

$$[a_1, b_1] = [0, OR] \cap [IR_0, IR_1],$$

$$[a_2, b_2] = [OR, IR_{max}] \cap [IR_0, IR_1].$$

La figure 4.6 montre la probabilité de réalisation de chaque état futur en fonction du nombre de connexions. Ces probabilités sont calculées en se basant sur les équations 4.35 (avec $D = 400 \text{ ms}$, $OR = 160 \text{ paquets}$, $IR_{max} = 300 \text{ paquets}$).

Lorsque le nombre de connexions est faible (la charge est faible) la probabilité qu'il n'y ait pas de future congestion et la probabilité de congestion légère proche (P_4 et P_2) sont importantes. Ces probabilités diminuent lorsque le nombre de connexions augmente.

D'autre part, les probabilités d'une future congestion grave ou légère lointaine (P_1 et P_3) sont faibles pour un faible nombre de connexions. Elles augmentent lorsque

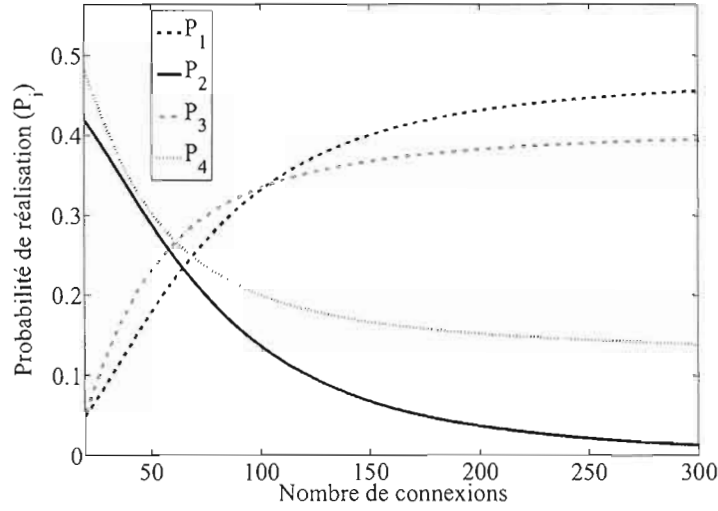


Figure 4.6 Probabilité de réalisation de chaque état futur en fonction du nombre de connexions.

ce nombre augmente. Ceci montre que les résultats analytiques sont conformes aux résultats attendus.

- Taille moyenne de la file d'attente

Soit Q_i la taille moyenne de la file d'attente au cours de l'état i . Par conséquent, la taille moyenne de la file Q_s est donnée par :

$$Q_s = \sum_{i=1}^4 P_i Q_i. \quad (4.39)$$

Soit IR_c le débit moyen lors d'une congestion (c.-à-d. $OR < IR(\tau) < IR_{max}$). Puisque $IR(\tau)$ est uniformément distribué, IR_c est donné par :

$$IR_c = OR + \frac{IR_{max} - OR}{2}. \quad (4.40)$$

Soit IR_p le débit moyen s'il n'y a pas de congestion (c.-à-d. $0 < IR(\tau) < OR$).

Puisque $IR(\tau)$ est uniformément distribué, IR_p est donné par :

$$IR_p = \frac{OR - 0}{2}. \quad (4.41)$$

Afin de simplifier la notation, on définit la fonction $(.)^+$ telle que $(x)^+ = x$ si $x > 0$ sinon 0.

Dans ce qui suit, nous estimons la taille moyenne de la file pour chaque état futur avec ou sans l'utilisation du mécanisme α -SNFAQM.

– *Sans α -SNFAQM* :

À la fin d'un intervalle τ , la taille de la file d'attente est égale à la différence entre le débit de sortie et le débit reçu.

Par exemple, à la fin d'une congestion grave, $IR_c - OR$ octets resteront dans la file à l'intervalle τ et de même pour l'intervalle $\tau + 1$. Ainsi, on a :

$$Q_1 = IR_c - OR. \quad (4.42)$$

Pour une future congestion légère proche, la taille de la file à la fin de l'intervalle τ est égale à $IR_c - OR$ et elle est égale à $IR_p - OR$ pour l'intervalle $\tau + 1$. Ainsi, on a :

$$Q_2 = \frac{((IR_c - OR) + (IR_p - OR))^+}{2}. \quad (4.43)$$

Pour une future congestion légère lointaine, la taille de la file à la fin de l'intervalle τ est nulle (puisque $IR(\tau) < OR$) et elle est égale à $IR_c - OR$ pour l'intervalle $\tau + 1$. Ainsi, on a :

$$Q_3 = \frac{IR_c - OR}{2}. \quad (4.44)$$

S'il n'y a pas de congestion future, la taille de la file d'attente à la fin des intervalles τ et $\tau + 1$ est nulle (puisque $IR(\tau) < OR$)

$$Q_4 = 0. \quad (4.45)$$

La taille moyenne de la file Q_s est ensuite calculée en utilisant l'équation 4.39.

– Avec α_SNFAQM :

Le mécanisme α_SNFAQM est activé lorsque la taille de la file dépasse th_{min} . Ainsi, la taille minimale de la file est th_{min} .

Pour une congestion grave, α_SNFAQM rejette systématiquement tous les paquets ($q(\tau) = 1$) en maintenant la taille de la file à th_{min} pendant les intervalles τ et $(\tau + 1)$. Par suite, la taille de la file d'attente à la fin d'une congestion grave est :

$$Q_1 = th_{min} \quad (4.46)$$

S'il y a une congestion légère lointaine, α_SNFAQM rejette les paquets avec une probabilité ϵ . Ainsi, pendant l'intervalle τ , le débit d'entrée moyen IR_p est inférieur à OR et par suite, la taille de la file est nulle puisque les IR_p octets vont être servis. Pendant l'intervalle $\tau + 1$, le débit d'entrée peut être estimé par $(1 - \epsilon)IR_C$ en considérant les rejets effectués par le mécanisme α_SNFAQM . Ainsi, la taille de la file à la fin d'une congestion légère lointaine est égale à :

$$Q_3 = \max(th_{min}, \frac{((1 - \epsilon)IR_C - OR)^+}{2}) \quad (4.47)$$

S'il n'y a pas de congestion ou s'il y a une congestion légère proche, le mécanisme accepte les paquets. Par suite, on a la même taille de la file que celle du cas sans α_SNFAQM (en ajoutant th_{min}). Ainsi, on a :

$$Q_2 = \max(th_{min}, \frac{((IR_c - OR) + (IR_p - OR))^+}{2}) \quad (4.48)$$

$$Q_4 = 0 \quad (4.49)$$

La taille moyenne de la file Q_s est calculée en utilisant l'équation 4.39.

La figure 4.7 montre la taille moyenne de la file d'attente avec et sans l'utilisation du mécanisme α _SNFAQM. Sans le mécanisme α _SNFAQM, lorsque le nombre de connexions augmente, la taille moyenne de la file d'attente augmente. Par contre, l'utilisation de ce mécanisme stabilise la taille moyenne de la file quel que soit le nombre de connexions.

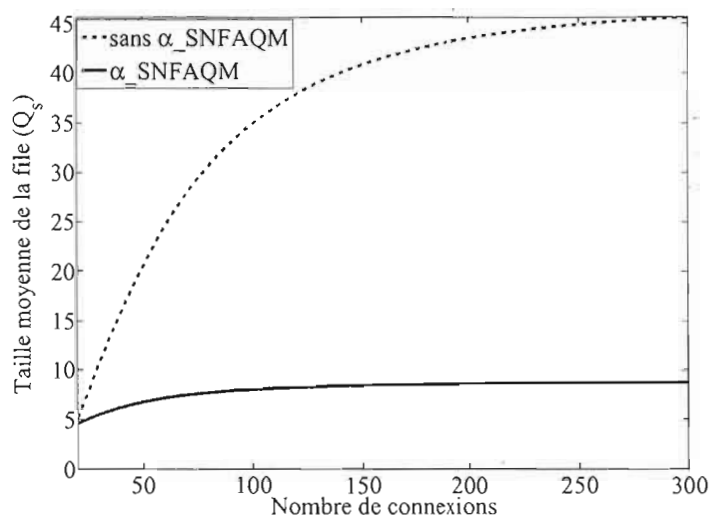


Figure 4.7 Taille de la file d'attente en fonction du nombre de connexions ($th_{min} = 5$).

4.3 Évaluation de performance

Nous comparons le mécanisme α _SNFAQM par rapport aux autres AQM tels que RED, PAQM et APACE. Nous avons choisi PAQM et APACE puisqu'ils se basent sur la prévision du trafic. De plus, les travaux précédents ont montré qu'ils sont plus performants que plusieurs autres mécanismes tels que RED, SRED et AVQ en termes de stabilité de la taille de la file et en termes d'utilisation de la file (Gao, He et Hou, 2002; Jain, Karandikar et Verma, 2003; Jain, Karandikar et Verma, 2004).

Pour les simulations, nous avons implémenté l'algorithme α _SNFAQM dans un module en C++ qui a été ensuite intégré au simulateur *Network Simulator 2* (ns-2).

(McCanne et Floyd, 1995). Deux scénarios différents ont été simulés pour effectuer les comparaisons entre les mécanismes étudiés.

4.3.1 Simulation du premier scénario

Topologie

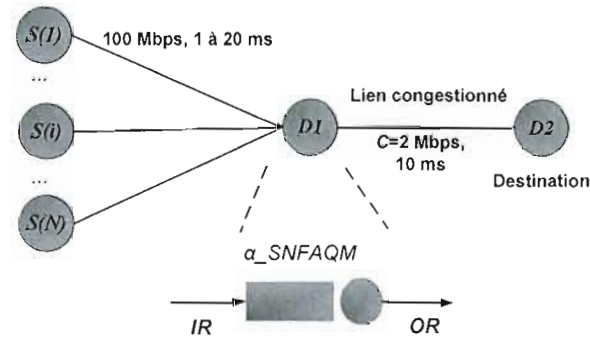


Figure 4.8 Topologie du premier scénario.

La figure 4.8 montre la topologie du réseau étudié. Il y a plusieurs sources $S(i)$. Chaque source génère 254 sessions TCP vers la destination $D2$. Ainsi, tout le trafic TCP passe par le même lien ($D1$ - $D2$) de capacité 2 Mbps.

Chaque session envoie des fichiers en utilisant le protocole FTP. La taille des fichiers suit la distribution de Pareto de paramètre $\alpha = 1,5$ et de valeur minimale $x_{min} = 10 \text{ MB}$ (sect. 1.15). La distribution de Pareto est utilisée puisqu'elle permet de générer un trafic auto-similaire (Leland et al., 1994; Park, Kim et Crovella, 1997; Charzinski, 2000; Downey, 2001). La taille d'un paquet est égale à 1000 octets. Le début de transfert des fichiers suit un processus de Poisson de paramètre 0,4 (25 connexions sont créées chaque seconde). La taille maximale de la file d'attente est fixée à 50 paquets.

Les mêmes simulations ont été effectuées pour chacun des mécanismes (RED, α -SNFAQM, APACE et PAQM). Les paramètres utilisés pour α -SNFAQM sont

$th_{min} = 4$, $th_{max} = 15$, $w_q = (0,002)$, $\epsilon = (0,8)$, $I = 180$ ms et deux retards comme variables d'entrée du modèle de prévision. Pour le mécanisme RED, les paramètres sont $max_p = (0,15)$, $th_{min} = 4$, $th_{max} = 15$ et $w_q = 0,002$ (app. B). Pour le mécanisme APACE, les paramètres sont $\alpha = (0,1)$, $max_p = (0,2)$ et 15 retards pour le modèle de prévision (éq. 1.47). Les paramètres de PAQM sont $I = 180$ ms, $\bar{Q} = 4$ et trois retards comme variables d'entrée pour le modèle de prévision (éq. 1.46).

Ces paramètres ont été choisis de sorte que l'objectif de chacun de ces mécanismes soit la stabilisation de la taille de la file autour de quatre paquets. Ceci rend possible la comparaison entre eux.

Les simulations ont été réalisées en variant à chaque fois le nombre de sources (de 3 à 30). Le temps de propagation entre une source $S(i)$ et le noeud $D1$ varie de 1 ms à 20 ms. Ceci signifie que le temps d'aller-retour (RTT) entre les sources et la destination $D2$ varie de 22 ms à 60 ms. Ceci rend les paramètres de la simulation plus réalistes.

La durée de chaque simulation est de 300 secondes. Cette durée est suffisante pour atteindre un état stationnaire pour le système.

La figure 4.9 montre comment la charge de la file d'attente (IR. moyen/OR) augmente quand le nombre de sources augmente. Plus cette charge augmente, plus la file d'attente est congestionnée.

Résultats

Les figures 4.10 et 4.11(a)) montrent la taille de la file d'attente pour le mécanisme α -SNFAQM et les états futurs prédits.

- Chaque fois qu'il y a une future congestion grave, la taille de la file d'attente reste autour de $th_{min} = 4$ paquets.

- Chaque fois qu'il y a une future congestion légère proche, α -SNFAQM accepte les paquets puisqu'il n'y a pas de congestion future. Ceci permet d'éviter les rejets non

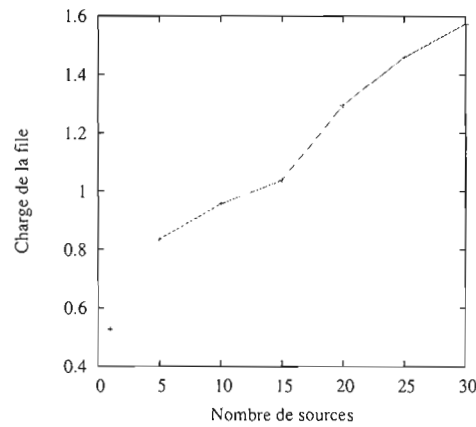


Figure 4.9 Charge de la file d'attente en fonction du nombre de sources.

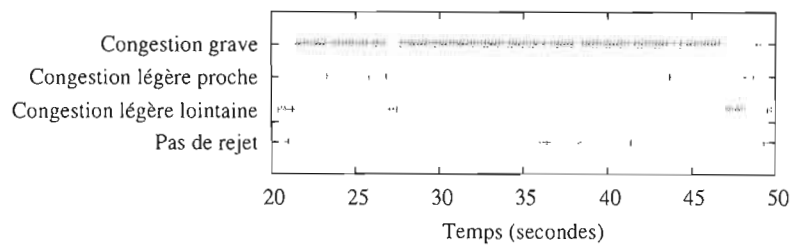
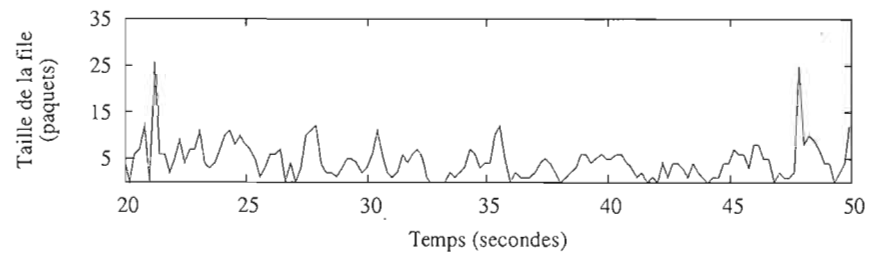
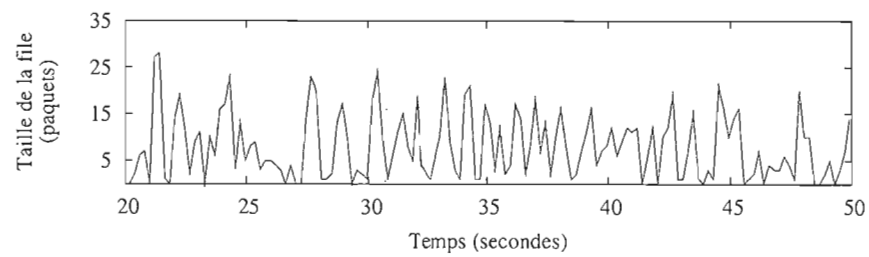


Figure 4.10 Les états futurs de la file d'attente prédits par le mécanisme α _SNFAQM (10 sources).

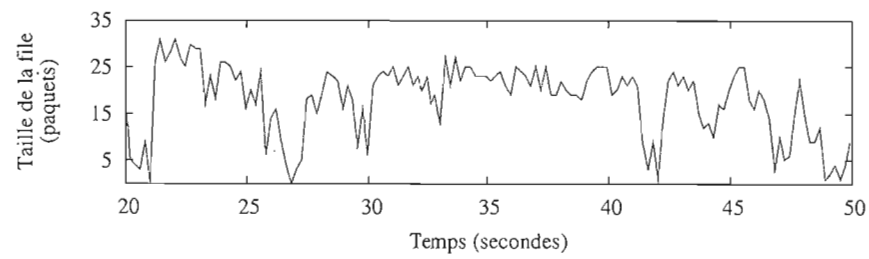
nécessaires de paquets et d'absorber la congestion actuelle.

- S'il n'y a ni congestion grave ni congestion légère, les paquets sont acceptés. La taille moyenne de la file sera comprise entre th_{min} et th_{max} .

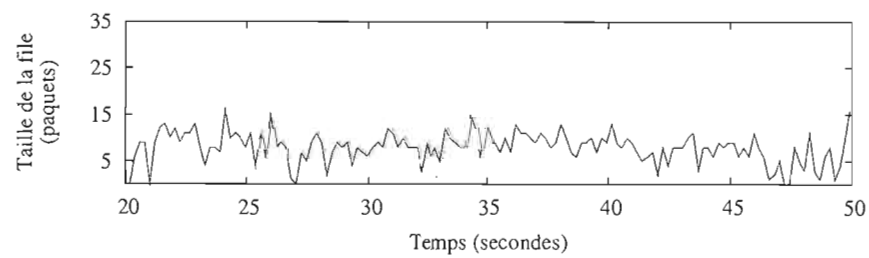
D'après la figure 4.11, on remarque que α _SNFAQM et PAQM réduisent la taille de la file par rapport à RED. La taille de la file est généralement proche du $th_{min} = 4$ pour le α _SNFAQM sauf dans le cas de congestion légère proche. On remarque que la taille de la file d'attente est en moyenne égale à quatre paquets pour le mécanisme α _SNFAQM tandis qu'elle est égale à 8 pour PAQM et RED et égale à 17 pour APACE.

(a) α -SNFAQM

(b) RED



(c) APACE



(d) PAQM

Figure 4.11 Taille de la file d'attente (10 sources).

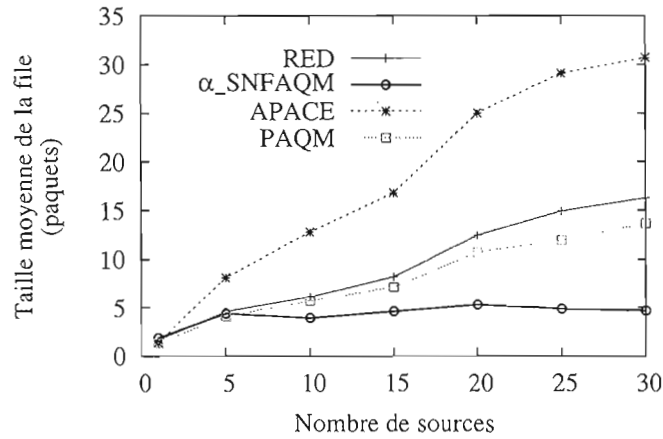


Figure 4.12 Taille moyenne de la file d'attente.

La figure 4.12 montre la taille moyenne de la file d'attente pour les différents mécanismes. Le nombre de sources a été varié de 1 à 30. On remarque que la taille moyenne de la file reste faible pour les mécanismes PAQM, RED et α -SNFAQM par rapport au mécanisme APACE. De plus, pour les mécanismes RED et PAQM, elle augmente lorsqu'on augmente le nombre de sources. Le mécanisme α -SNFAQM garantit non seulement une faible taille pour la file d'attente, mais aussi garantit que la valeur de celle-ci reste autour de th_{min} quelle que soit la charge de la file d'attente et notamment pour les charges importantes. Ceci est conforme aux résultats analytiques illustrés par la figure 4.7. Par conséquent, on peut contrôler la taille moyenne de la file d'attente en contrôlant le paramètre th_{min} . En d'autres termes, nous pouvons contrôler le délai moyen d'attente des paquets dans la file quelle que soit la quantité du trafic entrant.

La figure 4.13 montre le débit de perte pour les différents mécanismes AQM. α -SNFAQM rejette pratiquement la même quantité de données que les mécanismes APACE et PAQM. Le mécanisme RED rejette plus de paquets, notamment pour les grandes charges.

La figure 4.14 montre le débit de sortie de la file d'attente pour chaque simulation. Tous les mécanismes offrent le même débit de sortie, c'est-à-dire une même utilisation

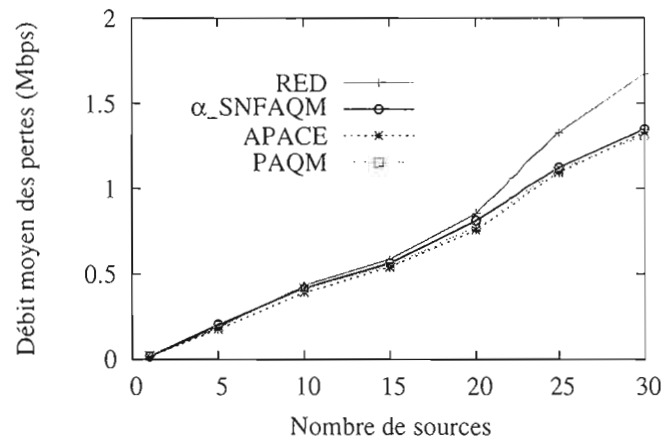


Figure 4.13 Débit moyen de perte des paquets.

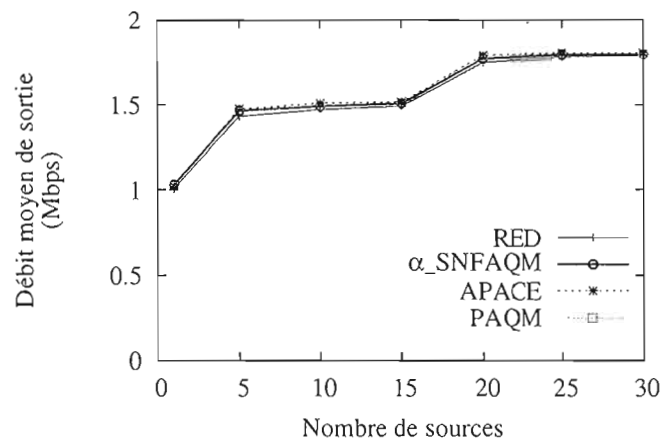


Figure 4.14 Débit moyen de sortie de la file d'attente.

du lien de sortie. On compare aussi les différents AQM en termes de débit à la réception (appelé *Goodput*), c.-à-d. le débit des données reçues au niveau de l'application sans compter les retransmissions et les acquittements TCP. Le débit à la réception (*Goodput*) est pratiquement le même pour tous les mécanismes (fig. 4.15)

Les résultats montrent que α _SNFAQM offre une meilleure performance par rapport aux autres mécanismes. En particulier, il permet de stabiliser la taille de la file

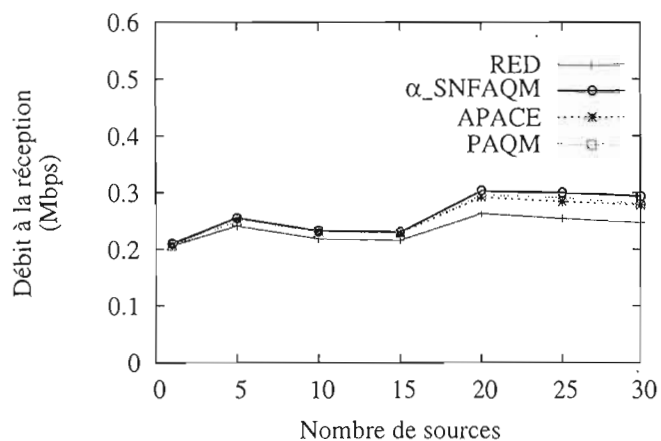


Figure 4.15 Débit à la réception (niveau application).

d'attente tout en gardant une utilisation élevée du lien.

4.3.2 Simulation du deuxième scénario

Topologie

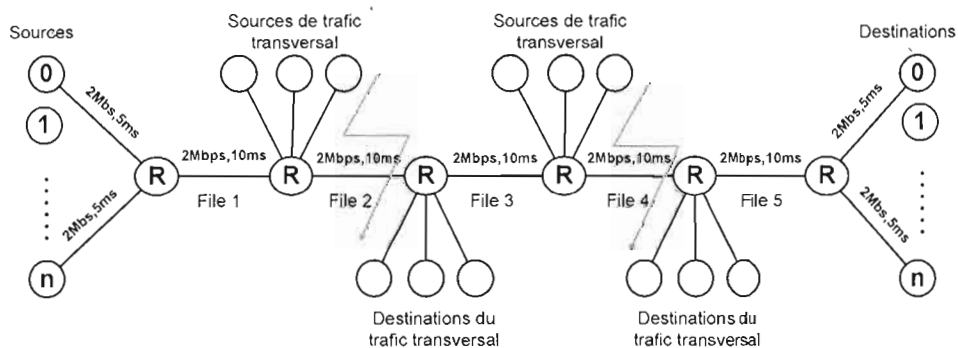


Figure 4.16 Topologie du deuxième scénario.

Dans cette section, nous comparons les différents mécanismes AQM dans une autre topologie. Contrairement au scénario 1, le trafic passe par une cascade de nœuds.

Ceux-ci sont aussi traversés par un trafic transversal provenant d'autres sources.

La figure 4.16 montre la topologie du réseau étudiée. Il y a six files d'attente qui sont traversées par le trafic des sources vers les destinations. Les deux files d'attente congestionnées (2 et 4) sont traversées par un trafic transversal. Il y a des connexions TCP qui sont établies entre chaque source (à gauche) et chaque destination (à droite). Chaque source TCP envoie des fichiers en utilisant le protocole FTP. Le début de transfert suit un processus de Poisson de paramètre 0,4 (25 connexions sont créées chaque seconde). La taille des fichiers suit la distribution de Pareto de paramètre 1,5 et de moyenne 10 MB. La distribution de Pareto est utilisée puisqu'elle permet de générer un trafic auto-similaire (Leland et al., 1994; Park, Kim et Crovella, 1997; Charzinski, 2000; Downey, 2001). La taille d'un paquet est égale à 1000 *octets*. L'intervalle de contrôle I pour le α -SNFAQM est égal à 400 ms ($\approx RTT$). Les paramètres des différents mécanismes AQM sont les mêmes que ceux utilisés dans le premier scénario (sect. 4.8).

Les files d'attente 2 et 4 sont les plus congestionnées et ont pratiquement le même comportement. Ainsi, nous présentons seulement les résultats pour la file d'attente 2.

Résultats

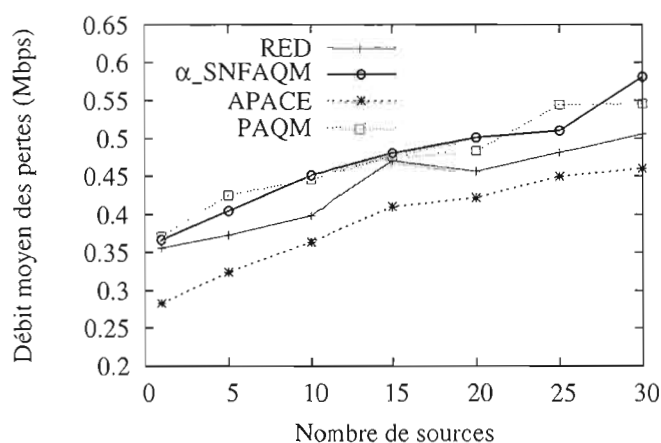


Figure 4.17 Débit moyen de la perte des paquets.

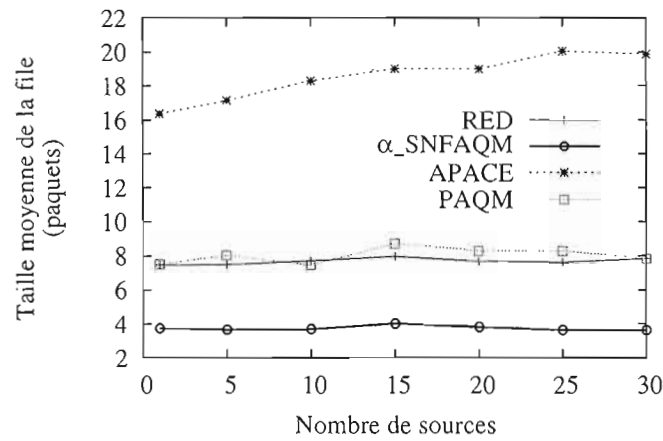


Figure 4.18 Taille moyenne de la file d'attente.

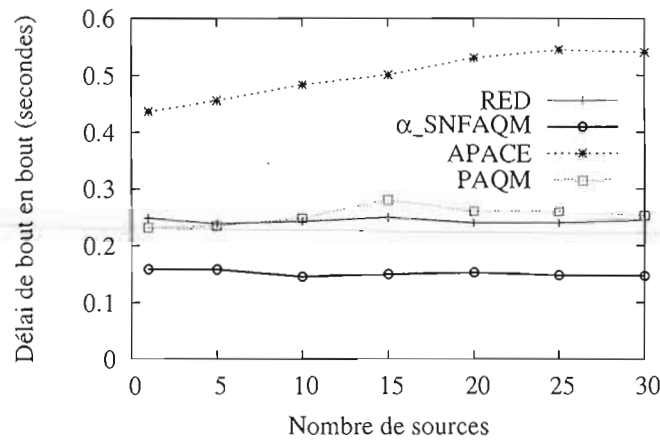
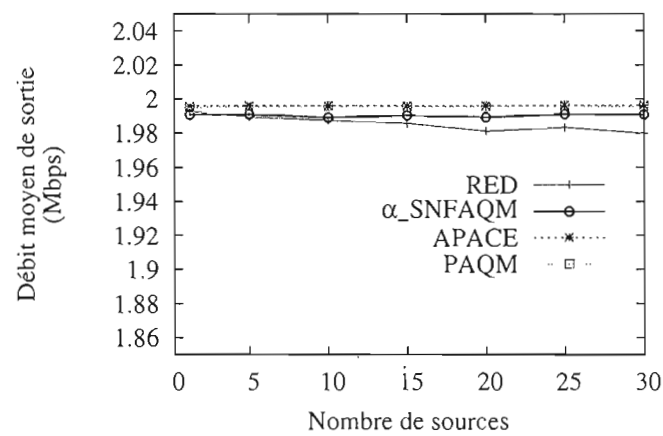


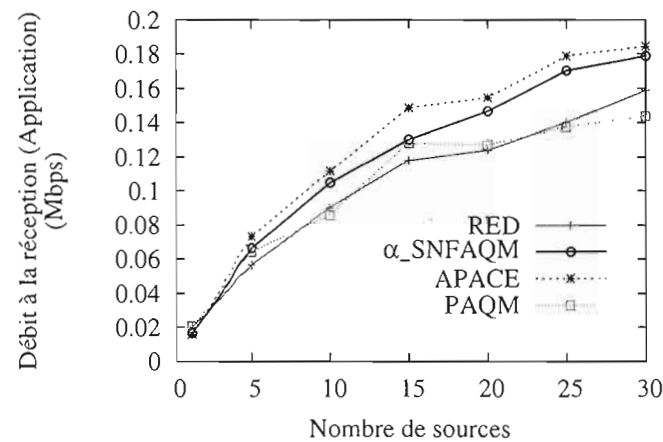
Figure 4.19 Délai moyen de bout en bout (entre la source 0 et la destination 0)

La figure 4.17 montre le débit de perte au niveau de la file d'attente 2 pour les mécanismes étudiés. On remarque que les mécanismes PAQM et α -SNFAQM ont plus de pertes comparés aux mécanismes APACE et RED. Cependant, le débit de perte augmente pour tous les mécanismes quand la charge du réseau augmente (c.-à-d. lorsque le nombre de sources augmente). La figure 4.18 montre la taille moyenne de la file lorsque le nombre de sources augmente. Contrairement au mécanisme APACE, les mécanismes

α -SNFAQM, PAQM et RED stabilisent la taille moyenne de la file quelle que soit la charge. Cependant l'avantage du α -SNFAQM est qu'il stabilise la taille autour de la valeur désirée th_{min} . Ainsi, il est capable de maintenir la stabilité de la taille moyenne de la file d'attente lorsque la charge du réseau change pour les deux scénarios étudiés. Cette stabilité se traduit par une stabilité du délai moyen de la transmission d'un paquet de bout en bout comme le montre la figure 4.19.



(a) Débit moyen de sortie (file 2).



(b) Débit à la réception (niveau application)

Figure 4.20 Débit moyen de sortie (file 2) et débit à la réception.

La figure 4.20(a) montre le débit moyen de sortie de la file d'attente 2. Le débit de sortie est presque le même pour tous les mécanismes. Il est égal à 2 Mbps (la capacité du lien). Il diminue légèrement avec le mécanisme RED pour les grandes charges. Bien que l'utilisation du lien soit importante pour tous les mécanismes, le débit à la réception au niveau de l'application est supérieur pour les mécanismes APACE et α -SNFAQM (fig. 4.20(b)). Cela veut dire que le mécanisme α -SNFAQM a engendré moins de retransmission de paquets TCP.

4.4 Conclusion

Dans ce chapitre, nous avons proposé un nouveau mécanisme de gestion de files d'attente, baptisé α -SNFAQM, qui se base sur la prévision du trafic. À l'aide du modèle de prévision α -SNF, le mécanisme proposé est capable de capturer les variations du trafic et de prédire les futures congestions. Ainsi, la technique permet de distinguer deux niveaux de congestion : congestion grave et congestion légère.

Le mécanisme est évalué par une étude analytique pour calculer la taille moyenne de la file d'attente avec l'hypothèse que les connexions TCP ont le même RTT et taux de perte dans un même intervalle de temps. L'analyse montre que le mécanisme α -SNFAQM stabilise la taille de la file d'attente autour d'une valeur th_{min} fixée par l'administrateur du réseau.

Nous avons évalué et comparé la performance du mécanisme α -SNFAQM avec celles des mécanismes PAQM, APACE et RED en termes de débit à la réception (goodput), de la taille moyenne de la file d'attente et du taux de perte de paquets. Les simulations ont été effectuées en utilisant une simple topologie et une topologie complexe comportant plusieurs liens congestionnés. Les résultats montrent que, pour les deux scénarios, le mécanisme α -SNFAQM offre une meilleure performance. En effet, il permet de stabiliser la taille de la file d'attente tout en gardant une utilisation élevée du lien. Il permet aussi de maximiser le débit à la réception (au niveau de la couche application) en minimisant les retransmissions TCP.

La taille de la file d'attente est maintenue autour d'une valeur qui peut être contrôlée (th_{min}). Cela permet de contrôler le délai de transmission des paquets de bout en bout. C'est un résultat très important en particulier pour la garantie de la qualité de service en termes de délai.

CHAPITRE V

APPLICATION DE LA PRÉVISION DU TRAFIC DANS LE CAS DES RÉSEAUX OBS

« Être réaliste, c'est préférer une réforme modeste, qui en permet une autre, à un miracle impossible. »

Habib Bourguiba

Nous avons vu dans le chapitre précédent que la modélisation et la prévision peuvent être utilisées pour prévenir les congestions dans le réseau et stabiliser la taille des files d'attente. Dans ce chapitre, la modélisation et la prévision du trafic sont appliquées pour contrôler le taux de perte dans les réseaux optiques à commutation de rafales (*Optical Burst Switched Networks* - OBS). Ainsi, nous proposons un mécanisme de contrôle d'admission utilisant la modélisation et la prévision afin de contrôler *le débit de burstification des rafales*, c'est-à-dire le débit avec lequel les rafales sont injectées dans le réseau. Cela permet de contrôler le taux de perte des rafales et de le stabiliser autour d'une valeur fixée par l'administrateur du réseau.

Dans la première section, nous présentons le modèle de système asservi en boucle fermée qui permet de contrôler le taux de perte dans le réseau en utilisant le modèle ARMA pour modéliser le réseau OBS. Nous proposons ensuite une généralisation du modèle pour implémenter la qualité de service (en définissant des classes de service). Nous proposons aussi une amélioration du modèle qui permet d'éviter les feedbacks du réseau en se basant sur l'estimation des paquets perdus au niveau de la couche TCP

pour prévoir le taux de perte du réseau.

La deuxième section est consacrée à l'évaluation de la performance du modèle avec feedback sans et avec qualité de service et du modèle sans feedback utilisant la prévision du trafic.

Dans ce qui suit, nous présentons le modèle proposé.

5.1 Modèle proposé pour le contrôle du taux de perte

Le modèle proposé pour contrôler le taux de perte dans le réseau OBS se base sur un contrôle d'admission qui est géré par un système asservi en boucle fermée. Un feedback du réseau, contenant le nombre de rafales perdues, est nécessaire pour contrôler le débit de burstification des rafales.

Le modèle a été aussi généralisé pour tenir compte de la qualité de service ; en d'autres termes, le modèle permettra de garantir un taux de perte pour chaque classe de service.

Le modèle est ensuite amélioré en ajoutant une technique permettant au nœud d'entrée d'estimer les pertes des rafales et de ne plus avoir besoin d'un feedback provenant du nœud destination. Ceci est effectué grâce à une prévision au niveau TCP et à une formule, que nous avons développée, qui permet de calculer les pertes au niveau OBS à partir des pertes au niveau de la couche TCP.

Dans ce qui suit, nous présentons le modèle de contrôle proposé, ensuite la modélisation du réseau OBS et la conception du contrôleur et enfin l'amélioration du modèle avec la prévision TCP.

5.1.1 Description du modèle de contrôle proposé

L'automatique est une discipline qui traite de la modélisation, de l'analyse, de la commande et, de la régulation des systèmes dynamiques. Elle a pour fondements

théoriques les mathématiques, la théorie du signal et l'informatique théorique. L'automatique permet de contrôler un système de manière continue et en respectant un cahier des charges (rapidité, stabilité, etc.)

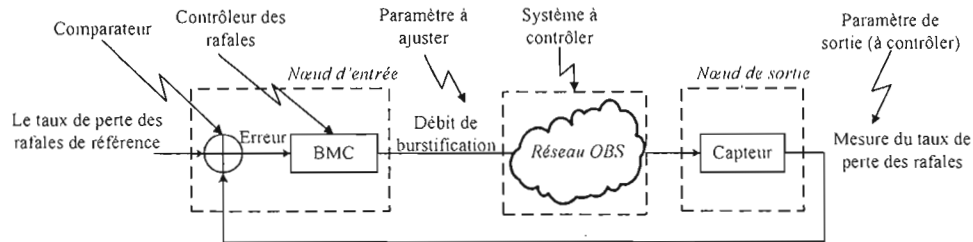


Figure 5.1 Modèle de système asservi en boucle fermée sans QoS.

Le modèle proposé se base sur ce qu'on appelle système asservi en boucle fermée. Une illustration de ses différentes composantes est présentée dans la figure 5.1.

- *Le nœud d'entrée* construit les rafales en assemblant un nombre de paquets IP dirigés vers une même destination (*nœud de sortie*). Cette opération s'appelle *la burstification*.
- *Le contrôleur de rafales* (*Burst Manager Controller - BMC*) est le composant du nœud d'entrée qui effectue la burstification. Ce composant effectue le contrôle : il utilise l'erreur comme entrée pour générer des rafales avec un débit de burstification selon une loi de contrôle.
- *Le débit de burstification* (*Burstification Rate - BR*) est le nombre de rafales envoyées par seconde par chaque BMC. Le débit de burstification est *le paramètre à ajuster* (*tuning parameter*) qui a une influence sur le comportement du système. Ce paramètre est ajusté dynamiquement pour contrôler le paramètre de sortie (à contrôler).
- *Le réseau OBS* est le système à contrôler.
- *Le taux de perte des rafales* (*Burst Loss Ratio - BLR*) est le paramètre de sortie qu'on veut contrôler et qui représente une métrique qui mesure la performance du réseau OBS.

- Le capteur (*Sensor*) lit la valeur mesurée du paramètre de sortie au niveau de chaque nœud de sortie c.-à-d. la mesure du taux de perte des rafales et renvoie la valeur comme feedback vers le nœud d'entrée.
- *Le taux de perte des rafales de référence* (*Reference Burst Loss Ratio - REF*) est la valeur « cible » que le paramètre de sortie (le taux de perte de rafales) doit avoir. Cette valeur est appelée aussi *consigne*. Ainsi, le contrôleur vise à stabiliser le taux de perte de rafales autour de cette valeur. Ce paramètre est fixé par l'administrateur du réseau.
- *L'erreur* est la différence entre la valeur du taux de perte de rafales de référence et la valeur du taux de perte des rafales mesurée et retournée par le capteur.

Comme le montre la figure 5.1, le contrôleur de rafales (BMC) au niveau du nœud d'entrée contrôle le débit de burstification. Un capteur situé au nœud de sortie renvoie au nœud d'entrée un feedback contenant le nombre de rafales reçues. Le nœud d'entrée calcule le taux de perte actuelle et l'erreur avec le taux de perte des rafales de référence.

Le BMC utilise cette erreur pour adapter le débit de burstification afin d'atteindre le taux de perte des rafales de référence.

La figure 5.2 montre le modèle proposé qui tient compte de tous les nœuds et de plusieurs classes de service (qualité de service).

Ce modèle a n nœuds d'entrée, m nœuds de sortie et w classes de service (fig. 5.2). Chaque nœud d'entrée s_i a un contrôleur de rafales ($BMC_{s_i d_j c_k}$) pour chaque nœud de sortie d_j et pour chaque classe de service c_k . Ainsi, il y a $m \times c$ contrôleurs de rafales au niveau de chaque nœud d'entrée. Chaque nœud de sortie d_j a un capteur ($Sensor_{s_i d_j c_k}$) pour chaque nœud d'entrée s_i et chaque classe de service c_k . Le capteur mesure périodiquement le taux de perte BLR pour chaque nœud d'entrée s_i , destination d_j et classe c_k ($BLR_{s_i d_j c_k}$) et renvoie la valeur comme feedback vers le contrôleur de rafales $BMC_{s_i d_j c_k}$ au niveau du nœud s_i .

La valeur de l'erreur ($E_{s_i d_j c_k}$) est calculée comme étant la différence entre le taux

de perte de référence ($REF_{s_i d_j c_k}$) et le taux de perte mesuré ($BLR_{s_i d_j c_k}$). Le contrôleur de rafale $BMC_{s_i d_j c_k}$ calcule le débit de burstification ($BR_{s_i d_j c_k}$) en se basant sur l'erreur. Le débit de burstification calculé est utilisé pour effectuer un contrôle d'admission pour les rafales entrant au réseau OBS. Ainsi, si la valeur du débit de burstification excède la valeur donnée par le BMC, aucune rafale ne sera injectée dans le réseau. Dans ce qui suit, nous présentons la modélisation du réseau OBS et la conception du contrôleur.

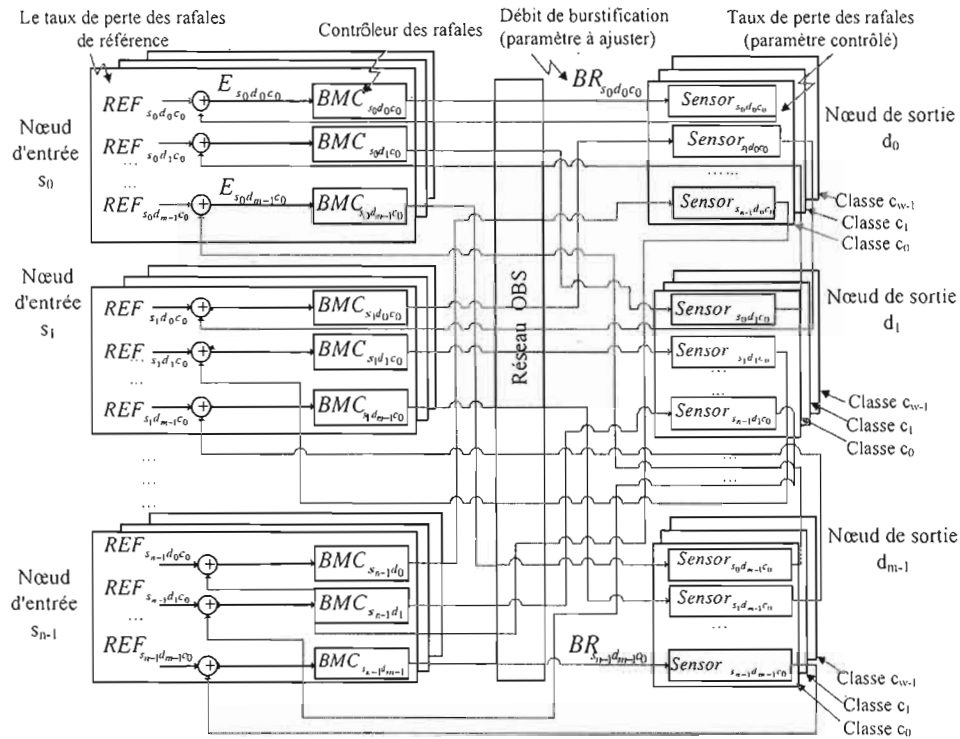


Figure 5.2 Modèle de système asservi en boucle fermée avec qualité de service.

5.1.2 Modélisation du réseau OBS et conception du contrôleur

La mise au point du contrôleur (BMC) est composée de deux phases. La première phase est *l'identification du système* qui consiste à construire une fonction de transfert qui relie les entrées du modèle à la sortie du modèle. La deuxième phase est *la conception du contrôleur* qui consiste à trouver les paramètres du contrôleur, notamment la loi de

contrôle.

– L'identification du système

Les contrôleurs utilisent la relation entre les entrées et les sorties; celle-ci est définie par un modèle mathématique. Le modèle ARMA est un modèle linéaire utilisé afin de relier les entrées et les sorties (Choi, 1992). En effet, bien que le contrôle non-linéaire peut être plus approprié pour les réseaux OBS, il est beaucoup plus difficile à appliquer (Aly, Zhan et Elbiaze, 2007b). La forme générale du modèle ARMA est donnée par :

$$y(t) = \sum_{p=1}^n a_p y(t-p) + \sum_{q=0}^m b_q x(t-q), \quad (5.1)$$

où $x(t)$, qui est l'entrée du modèle ARMA, représente le paramètre à régler. La sortie, $y(t)$, représente le paramètre contrôlé. Les paramètres n and m sont l'ordre du modèle. Les valeurs de a_p et b_q sont des constantes qui sont estimées à partir des mesures en utilisant la méthode des moindres carrés (Wonnacott et Wonnacott, 1990). La fonction de transfert du modèle peut être calculée après avoir identifié les valeurs de n , m , a_p et b_q . Le modèle ARMA est utilisé pour relier la sortie du modèle aux entrées ainsi qu'à l'historique de la sortie.

Nos simulations ont montré que le modèle ARMA modélise efficacement le réseau OBS entre une source s_i et la destination d_j pour une classe c_k (ces simulations sont présentées à la section 5.2). Ainsi, on considère que $y(t)$ est le taux de perte ($blr_{s_i d_j c_k}(t)$) des rafales de classe c_k qui partent de la source s_i vers la destination d_j et que l'entrée $x(t)$ est le débit de burstification ($br_{s_i d_j c_k}(t)$) des rafales de classe c_k partant de la source s_i vers la destination d_j . Les paramètres du modèle ARMA sont ($n = 1$ et $m = 0$). L'équation 5.1 devient :

$$blr_{s_i d_j c_k}(t) = a_1 blr_{s_i d_j c_k}(t-1) + b_0 br_{s_i d_j c_k}(t). \quad (5.2)$$

Les algorithmes de contrôle reposent sur la transformée de Laplace pour les systèmes à temps continu et la transformée en Z pour ceux à temps discret. Le modèle

proposé est défini dans le domaine de temps discret. Par conséquent, la transformée en Z est appliquée.

$$Y(z) = \sum_{t=0}^{\infty} y(t)z^{-t}. \quad (5.3)$$

L'équation 5.3 représente $Y(z)$, la transformée en Z de $y(t)$, où z est un nombre complexe (Ogata, 1997). Ceci permet d'utiliser les principes existants de l'automatique qui sont généralement basés sur le domaine des fréquences.

Pour la notation, on utilise les caractères minuscules pour les variables dans le domaine temporel, (par exemple, $y(t)$). Pour le domaine fréquentiel, les majuscules sont utilisées (par exemple, $Y(z)$).

En appliquant ces principes à l'équation (5.1), nous obtenons la formulation du modèle ARMA dans le domaine fréquentiel :

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{j=0}^m b_j z^{n-j}}{z^n - (\sum_{i=1}^n a_i z^{n-i})}. \quad (5.4)$$

En appliquant la même transformation à l'équation (5.2), on obtient :

$$OBS_{s_i d_j c_k}(z) = \frac{BLR_{s_i d_j c_k}(z)}{BR_{s_i d_j c_k}(z)} = \frac{z b_0}{z - a_1}, \quad (5.5)$$

où $OBS_{s_i d_j c_k}(z)$ modélise le réseau pour chaque nœud d'entrée (s_i) et chaque nœud de sortie (d_j) et la classe c_k des rafales. Cette équation sera utilisée, dans ce qui suit, pour concevoir le contrôleur.

- Conception du contrôleur

La loi de contrôle décrit la manière avec laquelle le contrôleur change la valeur du paramètre à régler. Le contrôleur produit une action qui continue à ajuster et corriger le paramètre à régler tant que l'erreur n'est pas nulle. Si l'erreur est faible, le contrôleur intégral continue à ajuster lentement. Si l'erreur est importante, le contrôleur ajuste plus rapidement le paramètre à régler.

Nous avons utilisé la loi de contrôle intégrale pour sa simplicité (Ogata, 1997). Le contrôleur intégral a la formule générale suivante dans le domaine temporel :

$$br_{s_i d_j c_k}(t) = br_{s_i d_j c_k}(t-1) + K_{s_i d_j c_k} e_{s_i d_j c_k}(t-1), \quad (5.6)$$

où $br_{s_i d_j c_k}(t)$ est le débit de burstification des rafales de classe c_k partantes du nœud d'entrée s_i vers le nœud de sortie d_j . Le réel $K_{s_i d_j c_k}$ est le gain associé à chaque contrôleur de rafales et $e_{s_i d_j c_k}(t)$ est la valeur de l'erreur entre le taux de perte des rafales et le taux de perte des rafales de référence.

La loi de contrôle fournit le débit de burstification maximale des rafales qui sont injectées dans le réseau OBS. Cette valeur est ajustée dynamiquement en utilisant l'ancienne mesure du taux de perte de rafales et l'erreur de contrôle correspondante. Lorsque la valeur du gain $K_{s_i d_j c_k}$ devient importante, la valeur du débit de burstification varie plus rapidement. Cependant, une valeur importante du gain $K_{s_i d_j c_k}$ (gain agressif) peut causer des oscillations ou aussi une instabilité.

Ainsi, dans ce qui suit, nous nous proposons de chercher le gain qui permet d'assurer la meilleure stabilité et performance pour le système.

On commence par calculer la fonction de transfert du modèle à boucle fermée ($T_{s_i d_j c_k}(z)$) pour le système en fonction de z (plus de détails sur la définition des fonctions de transfert pour un modèle à boucle fermée sont fournies dans l'appendice C). Ensuite, la technique de *la représentation des lieux des pôles (root locus)* permet de calculer le gain $K_{s_i d_j c_k}$ en se basant sur ($T_{s_i d_j c_k}(z)$). La figure 5.3 montre les différents composants du modèle avec les transformées en Z des différentes fonctions. Ainsi, on a :

$$T_{s_i d_j c_k}(z) = \frac{BLR_{s_i d_j c_k}(z)}{REF_{s_i d_j c_k}(z)} \quad (5.7)$$

En utilisant les propriétés de la transformée en Z , les transformées en Z de $br_{s_i d_j c_k}(t)$ et $br_{s_i d_j c_k}(t-1)$ sont respectivement $BR_{s_i d_j c_k}(z)$ et $\frac{1}{z}BR_{s_i d_j c_k}(z)$. Ainsi, l'application de l'équation (5.3) à la formule générale du contrôleur (Équation (5.6)) donne :

$$BR_{s_i d_j c_k}(z) = \frac{1}{z}BR_{s_i d_j c_k}(z) + K_{s_i d_j c_k} \frac{1}{z}E_{s_i d_j c_k}(z), \quad (5.8)$$

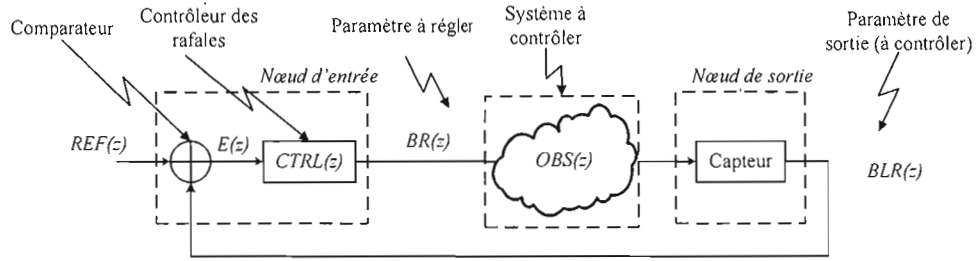


Figure 5.3 Modèle de système asservi en boucle fermée - les transformées en Z .

où

$$E_{s_i d_j c_k}(z) = REF_{s_i d_j c_k}(z) - BLR_{s_i d_j c_k}(z). \quad (5.9)$$

En utilisant $BR_{s_i d_j c_k}(z)$ comme facteur dans l'équation (5.8), on obtient :

$$BR_{s_i d_j c_k}(z) \left(1 - \frac{1}{z}\right) = K_{s_i d_j c_k} \frac{1}{z} E_{s_i d_j c_k}(z). \quad (5.10)$$

La forme générale de la fonction de transfert du contrôleur, $CTRL_{s_i d_j c_k}(z)$, est donnée par :

$$CTRL_{s_i d_j c_k}(z) = \frac{BR_{s_i d_j c_k}(z)}{E_{s_i d_j c_k}(z)} = K_{s_i d_j c_k} \left(\frac{1}{z}\right) \left(\frac{z}{z-1}\right). \quad (5.11)$$

Ainsi, on a :

$$CTRL_{s_i d_j c_k}(z) = \frac{BR_{s_i d_j c_k}(z)}{E_{s_i d_j c_k}(z)} = K_{s_i d_j c_k} \frac{1}{z-1}. \quad (5.12)$$

La forme générale de la fonction de transfert de $BLR_{s_i d_j c_k}(z)$ est donnée par :

$$BLR_{s_i d_j c_k}(z) = E_{s_i d_j c_k}(z) \frac{BR_{s_i d_j c_k}(z)}{E_{s_i d_j c_k}(z)} \frac{BLR_{s_i d_j c_k}(z)}{BR_{s_i d_j c_k}(z)}. \quad (5.13)$$

En substituant les équations (5.5) et (5.12) dans l'équation (5.13), on obtient :

$$BLR_{s_i d_j c_k}(z) = E_{s_i d_j c_k}(z) \left(K_{s_i d_j c_k} \frac{1}{z-1}\right) \frac{zb_0}{z-a_1}. \quad (5.14)$$

L'équation 5.14 constitue la fonction de transfert du modèle à boucle ouverte. En utilisant les équations (5.14 et 5.9), la fonction de transfert du modèle en boucle fermée ($T_{s_i d_j}(z)$) pour le système décrit par la figure 5.2 est donnée par :

$$T_{s_i d_j c_k}(z) = \frac{BLR_{s_i d_j c_k}(z)}{REF_{s_i d_j c_k}(z)} = \frac{K_{s_i d_j c_k} z b_0}{(z-1)(z-a_1) - K_{s_i d_j c_k} z b_0}. \quad (5.15)$$

En utilisant l'équation 5.15, la technique de *la représentation des lieux des pôles* (*root locus*) permet de calculer le gain $K_{s_i d_j c_k}$. Plus de détails sur le choix du gain adéquat et sur l'application de cette technique sont disponibles dans l'appendice D.

Ainsi, le gain $K_{s_i d_j c_k}$ peut être défini pour chaque triplet (source s_i , destination d_j , classe c_k). Connaissant la valeur du gain, le contrôleur peut être utilisé efficacement pour le contrôle du taux de perte dans le réseau.

5.1.3 Amélioration du modèle avec la prévision TCP

L'inconvénient principal du modèle de contrôle proposé (fig. 5.1) est le feedback qui est envoyé par le capteur situé à la destination. Ce feedback sert à informer périodiquement le nœud d'entrée du nombre de rafales reçus. Cette information est utilisée pour calculer le taux de perte des rafales ($BLR_{s_i d_j}$) partantes de s_i vers la destination d_j .

L'amélioration proposée consiste à remplacer le module du capteur situé à la destination par un module prédicteur situé à la source (fig. 5.4) (Zhani, Fouad et Elbiaze, 2009b). Le modèle de prévision estime le taux de perte des rafales $BLR_{s_i d_j}$ en se basant sur les statistiques disponibles au niveau de la couche TCP grâce à une méthode qu'on propose. Ainsi, il n'est plus nécessaire d'avoir un feedback provenant du réseau. Le contrôle du taux de burstification sera totalement effectué au niveau du nœud d'entrée.

Dans ce qui suit, nous décrivons le fonctionnement du module prédicteur proposé (fig. 5.4). Le module effectue la prévision en deux étapes :

1. L'estimation du débit de perte des segments au niveau TCP.

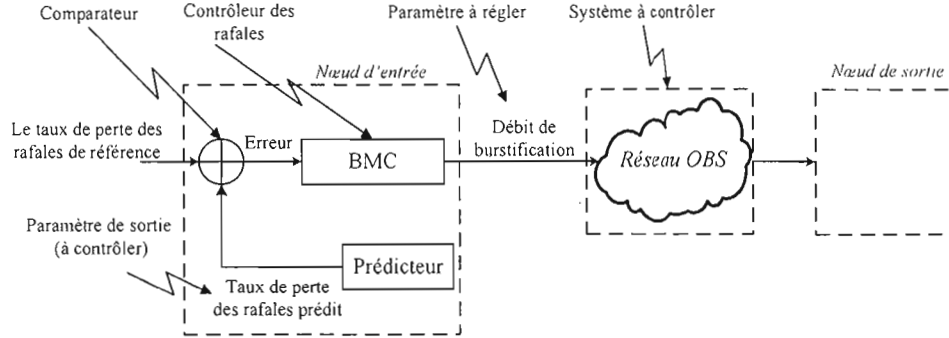


Figure 5.4 Modèle proposé sans feedback.

2. L'estimation du débit de perte des rafales au niveau OBS.

Avant de détailler ces deux étapes, nous présentons les notations utilisées pour les différentes variables.

La variable $BR_{s_i d_j}(\tau)$ est le débit des rafales partantes de la source s_i vers la destination d_j pendant l'intervalle de temps τ de taille I (c.-à-d. l'intervalle $[\tau, \tau + I]$). Afin de simplifier, le symbole τ ne sera pas précisé, ainsi on a :

- BR : débit des rafales (*Burst Rate*) c.-à-d. le nombre de rafales envoyées par intervalle de temps.
- SR : débit des segments (*Segment Rate*) c.-à-d. le nombre de segments TCP envoyés par intervalle de temps.
- BL : débit de la perte des rafales (*Burst Loss Rate*) c.-à-d. le nombre de rafales perdues par intervalle de temps.
- SL : débit de la perte des segments (*Segment Loss Rate*) c.-à-d. le nombre de segments TCP perdus par intervalle de temps.
- BLR : taux de perte des rafales (*Burst Loss Ratio*) donné par :

$$BLR = \frac{BL}{BR}. \quad (5.16)$$

- SLR : taux de perte des segments TCP (*Segment Loss Ratio*) donné par :

$$SLR = \frac{SL}{SR}. \quad (5.17)$$

- SpB : nombre de segments par rafale (*Segment Per Burst*) donné par :

$$SpB = \frac{SR}{BR}. \quad (5.18)$$

- L_v : probabilité de perte d'une rafale de taille v segments TCP.

Parmi ces variables, BR , SR , SpB peuvent être mesurées facilement au niveau du nœud d'entrée sans avoir besoin d'un feedback du réseau. Notre objectif est d'estimer les taux de perte SLR et BLR en utilisant seulement les informations locales disponibles au niveau du nœud. Par conséquent, le nouveau modèle n'utilisera plus un feedback du réseau. L'idée est d'utiliser les informations disponibles au niveau TCP sur les retransmissions et les pertes TCP afin d'estimer les pertes des rafales au niveau de la couche OBS.

Nous commençons par estimer le débit de perte des segments (*Estimated Segment Loss - ESL*). Ensuite, nous calculons l'estimation du débit de perte des rafales (*Estimated Burst Loss - EBL*).

- Estimation du débit de perte des segments (ESL)

La première étape est d'estimer le nombre de segments perdus noté par ESL . On suppose que la couche OBS est consciente de la couche TCP. Ainsi, la couche OBS peut détecter si un segment (reçu des couches supérieures) est un segment retransmis ou transmis pour la première fois. Cette hypothèse est réaliste et simple à implémenter, il suffit d'ajouter un drapeau dans les options du segment TCP qui permet de différencier les segments transmis pour la première fois et les segments retransmis. Ainsi, la couche OBS est capable d'évaluer le nombre de segments retransmis par intervalle de temps. Soit $RTS(\tau)$ le nombre de segments retransmis pendant l'intervalle de temps τ . On suppose que le nombre de segments perdus pendant l'intervalle $(\tau - 1)$ est égale au nombre de segment retransmis pendant l'intervalle τ . Par conséquent, le nombre de segments perdus $ESL(\tau)$ est estimé par :

$$ESL(\tau - 1) = RTS(\tau). \quad (5.19)$$

Afin de prévoir la valeur actuelle de $ESL(\tau)$, c'est-à-dire à l'intervalle τ , nous utilisons le modèle ARMA décrit à la section 1.5.1. Dans notre cas, $y(t)$ est $ESL(\tau)$. Afin de simplifier, les paramètres p et q sont fixés à 1. Les paramètres ϕ_1 et θ_1 sont estimés à partir des mesures collectées grâce à la *méthode des moindres carrés* (Wonnacott et Wonnacott, 1990). Les simulations présentées à la section 5.2.3 montrent que ces paramètres sont assez suffisants pour obtenir des prévisions précises. Ainsi, l'équation 1.21 devient :

$$ESL(\tau) = \phi_1 ESL(\tau - 1) + \epsilon(t) + \theta_1 \epsilon(t - 1). \quad (5.20)$$

Toutes les variables sont mises à jour chaque intervalle de temps τ de taille I . Idéalement, la valeur de I est un multiple du délai de retransmission (*Retransmission Timeout* - RTO) puisque les paquets sont retransmis chaque RTO.

– Comparaison avec les travaux précédents

Les travaux précédents sur l'estimation du nombre de segments TCP perdus ont trouvé que le nombre de retransmissions ne donne pas une bonne estimation du nombre de segments perdus (Allman, Eddy et Ostermann, 2003; Hoe, 1996; Floyd et Henderson, 1999). En effet, ils ont trouvé que la différence entre le nombre de paquet retransmis et le nombre de paquets perdu pour TCP (version Reno ou Tahoe) est due à l'utilisation de l'algorithme « *slow start* » lorsque le délai de retransmission (RTO) expire. En effet, fréquemment, le RTO expire et oblige TCP à retransmettre le paquet bien que ce dernier ne soit pas réellement perdu (Allman, Eddy et Ostermann, 2003; Hoe, 1996; Floyd et Henderson, 1999).

Nous pensons que ceci n'est pas vrai pour un réseau OBS. La figure 5.5.(a) montre un exemple de segment (Seg 1) qui a été perdu dans un réseau optique ordinaire (à commutation de paquet ou de circuit). Le segment est retransmis après le délai de retransmission (*Retransmission TimeOut* - RTO). Dans le cas où TCP n'utilise pas les rejets sélectifs, TCP retransmet les segments 2 et 3 même s'ils ne sont pas perdus. Ainsi, le nombre de segments perdus est inférieur à celui des paquets retransmis ($SL < RTS$).

Dans le cas où TCP utilise les rejets sélectifs, TCP ne retransmet que le segment 1. Ainsi, le nombre de segments perdus est presque égal à celui des paquets retransmis ($SL \approx RTS$).

La figure 5.5.(b) montre un autre exemple où le segment 1 est perdu dans un réseau OBS. Les segments qui le suivent sont aussi perdus puisqu'ils appartiennent à la même rafale (avec ou sans le rejet sélectif de TCP). Ainsi, le nombre de segments perdus est quasiment égal au nombre de segments retransmis ($SL \approx RTS$).

Le nombre de segments envoyés par une connexion TCP avant la réception d'un acquittement est égale à la taille de la fenêtre TCP (notée W). Ce nombre est généralement très petit par rapport à la taille de la rafale (Par exemple, nos simulations montrent que la taille moyenne de la fenêtre (10 segments) est faible par rapport à la taille d'une rafale (77 segments)). Ceci augmente la probabilité que les W segments envoyés par une connexion appartiennent à la même rafale. Par conséquent, ceci augmente la probabilité d'avoir le cas de la figure 5.5.(b) c'est-à-dire $SL \approx RTS$.

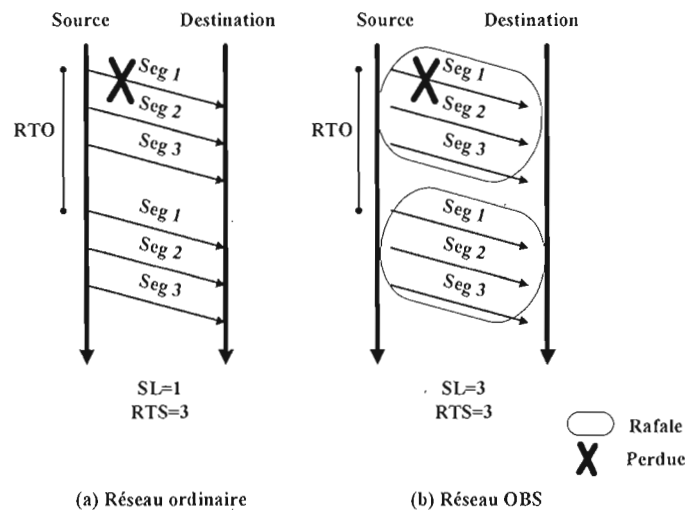


Figure 5.5 Exemple de retransmission TCP dans un réseau ordinaire et dans un réseau OBS.

Ceci montre que, dans le cas de la commutation de rafales, le nombre de segments retransmis donne une bonne estimation du nombre de segments perdus.

– Estimation du débit de perte des rafales (EBL)

Une fois que le nombre de segments perdus ESL a été estimé, il faut déduire EBL à partir de ESL . Ensuite, il est facile d'écrire $EBLR$ en fonction $ESLR$.

Puisque $L_v.BL$ est le nombre de rafales perdues de taille v segments, alors $v.(L_v.BL)$ est le nombre de segments dans les $(L_v.BL)$ rafales. Par suite, $v.(L_v.BL)$ est le nombre de segments perdus dans les rafales de taille v .

Par conséquent, le débit de perte des segments est exprimé par :

$$SL = \sum_{v=v_{min}}^{v_{max}} v.(L_v.BL), \quad (5.21)$$

où v_{min} et v_{max} sont respectivement les tailles minimale et maximale des rafales exprimées en nombre de segments. Ainsi, on a :

$$SL = BL. \sum_{v=v_{min}}^{v_{max}} v.L_v. \quad (5.22)$$

En divisant par BR et en utilisant l'équation 5.16, on a :

$$\frac{SL}{BR} = BLR. \sum_{v=v_{min}}^{v_{max}} v.L_v. \quad (5.23)$$

En utilisant l'équation 5.18, on a :

$$\frac{SL.SpB}{SR} = BLR. \sum_{v=v_{min}}^{v_{max}} v.L_v. \quad (5.24)$$

En utilisant l'équation 5.17, on obtient

$$SLR = BLR. \frac{\sum_{v=v_{min}}^{v_{max}} v.L_v}{SpB}. \quad (5.25)$$

Ensuite, on a :

$$BLR = \alpha.SLR. \quad (5.26)$$

où α est défini par :

$$\alpha = \frac{SpB}{\sum_{v=v_{min}}^{v_{max}} v.L_v}. \quad (5.27)$$

Ainsi, *EBLR* peut être déduit en fonction de *ESLR* par cette relation :

$$EBLR = \alpha.ESLR. \quad (5.28)$$

5.2 Évaluation de performance

Afin d'évaluer la performance du modèle proposé, nous l'avons implémenté dans le simulateur (ns-2) (McCanne et Floyd, 1995). Ainsi, nous validons le modèle avec feedback sur une topologie à deux nœuds de cœur sans qualité de service. Ensuite, le modèle avec feedback est appliqué à la topologie NSFNet avec trois classes de service afin de valider l'extensibilité du modèle pour des topologies complexes.

Finalement, nous évaluons le modèle amélioré (utilisant la prévision TCP) sans feedback sur la topologie NSFNet.

5.2.1 Contrôle du taux de perte avec feedback et sans QoS

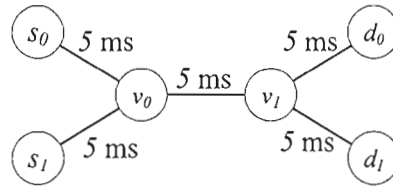


Figure 5.6 Topologie étudiée.

La topologie utilisée pour tester l'efficacité du système asservi en boucle fermée est constituée de deux nœuds d'entrée (les sources), deux nœuds de cœur et deux nœuds de sortie (destinations) comme le montre la figure 5.6 (Farahmand, Zhang et Jue, 2004). La configuration de la simulation est la suivante :

- Chaque nœud d'entrée génère des rafales suivant la distribution de Poisson de moyenne 7500 rafales par seconde.

- La taille d'une rafale est de 120 KB.
- Les taux de perte des rafales sont calculés à partir de la simulation comme un paramètre de bout en bout (source-destination).
- Le délai de traitement d'un paquet de contrôle est de $10 \mu s$.
- La valeur de l'offset est égale à 1 ms.
- Il n'y a pas de conversion de longueurs d'onde au niveau des nœuds.
- Il n'y a pas de mémoire tampon (buffer) pour conserver les rafales au niveau des nœuds.
- Pour le routage, l'algorithme du plus court chemin est utilisé.
- Chaque lien contient quatre canaux, chacun opère à 10 Gbps. Un seul canal est utilisé comme canal de contrôle, les canaux restants sont utilisés comme canaux de données.
- Le mécanisme de réservation est le protocole *Just-Enough-Time* (JET).
- Le taux perte des rafales de référence est fixé à 0,3.
- La technique de la représentation des lieux des pôles donne un gain $K_{s_i d_j}$ égal à 0,5.

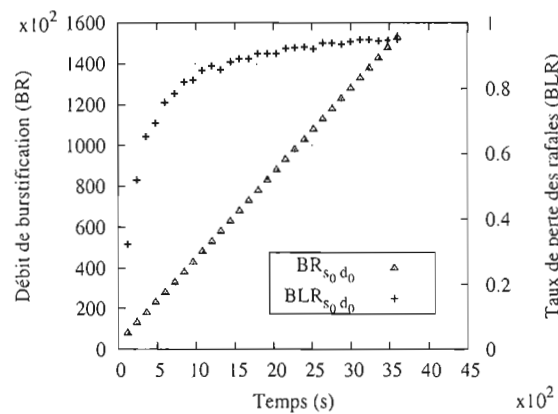


Figure 5.7 Débit de burstification et taux de perte des rafales entre la source s_0 et la destination d_0 .

La première étape est de tester s'il est possible de modéliser le réseau OBS par un modèle ARMA en considérant le débit de burstification (BR) comme entrée du modèle et le taux de perte des rafales (BLR) comme sortie. La figure 5.7 montre la relation entre le débit de burstification (BR) et le taux de perte des rafales (BLR). En effet, elle montre que le taux de perte des rafales augmente graduellement lorsque le débit de burstification augmente. L'impact du débit de burstification est clair ; ceci suggère que le débit de burstification sera suffisant comme entrée du modèle ARMA représentant le réseau OBS. Ainsi, il est facile de déduire à partir de ces données les paramètres du modèle ARMA (équ. 5.1) qui représente le réseau OBS et qui relie BLR à BR .

La figure 5.8 montre les taux de perte des rafales (le paramètre à contrôler) entre chaque noeud d'entrée et chaque noeud de sortie. Par exemple, la figure 5.8(a) montre

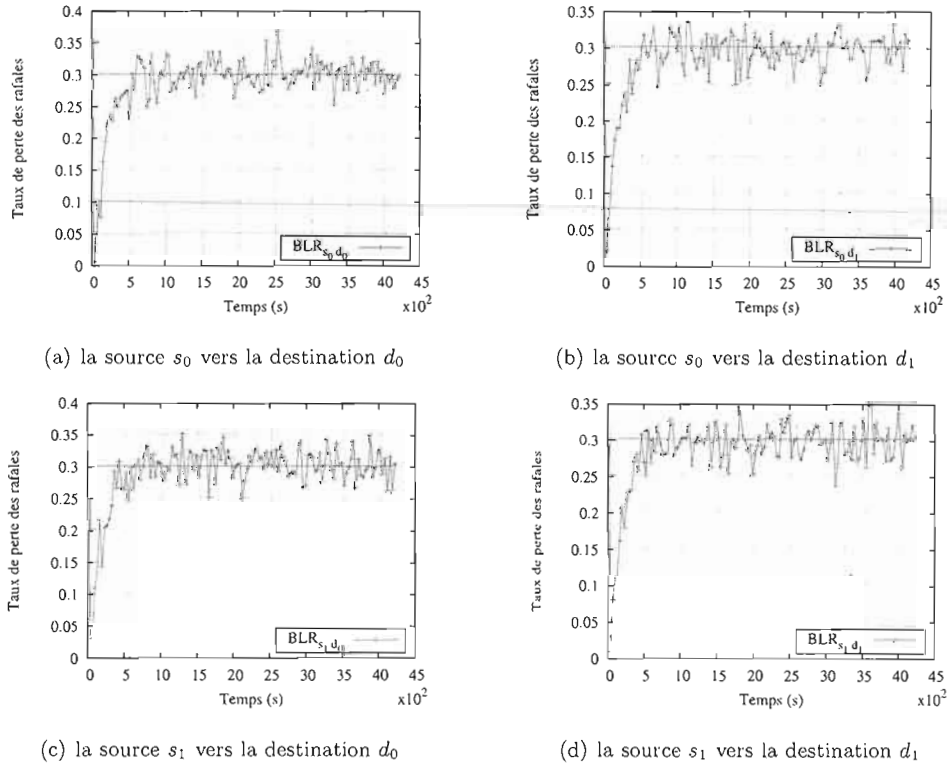


Figure 5.8 Taux de perte des rafales.

le taux de perte entre la source s_0 vers la destination d_0 . On remarque que le système se stabilise en moins de cinq seconde. Il est clair que le taux de perte des rafales oscille autour de la valeur de référence désirée 0,3 ($\pm 15\%$ de la référence). Ceci prouve que le contrôleur adapte efficacement le débit de burstification et par suite, le taux de perte désiré est atteint. Les résultats sont similaires pour toutes les sources et destinations.

La figure 5.9 montre le débit de burstification (le paramètre à régler) entre chaque noeud d'entrée et chaque noeud de sortie. Par exemple, la figure 5.9(a) montre le débit de burstification entre la source s_0 et la destination d_0 . En examinant aussi la figure 5.8(a), on remarque que lorsque la valeur du taux de perte dépasse 30 %, le contrôleur réduit le débit de burstification. Si la valeur du taux de perte est inférieure au taux désiré, le contrôleur augmente le débit de burstification. Le résultat est similaire pour toutes les

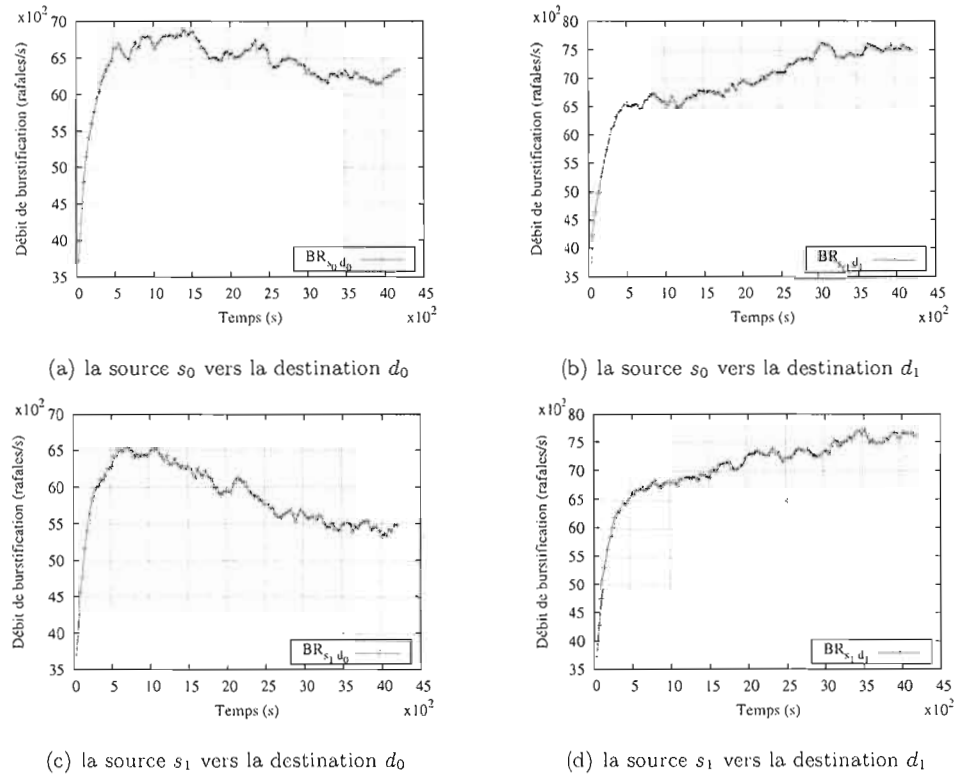


Figure 5.9 Débit de burstification.

sources et destinations (fig. 5.9). Cela montre que le contrôleur adapte efficacement le débit de burstification et par suite, le taux de perte désiré est atteint.

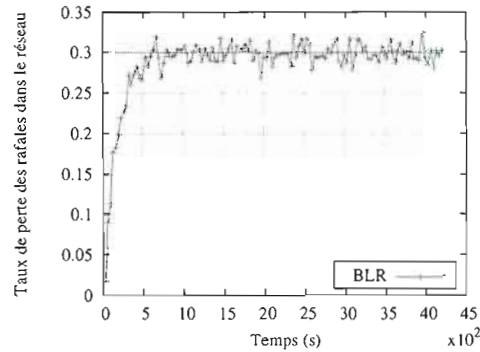


Figure 5.10 Taux de perte des rafales pour tout le réseau.

La figure 5.10 montre le taux de perte moyen des rafales pour tout le réseau durant la simulation. Le taux de perte oscille autour de la valeur de référence désirée (30 %). Ceci montre qu'on peut contrôler et de stabiliser le taux de perte des rafales dans le réseau autour d'une valeur fixée par l'administrateur du réseau grâce au modèle de contrôle proposé.

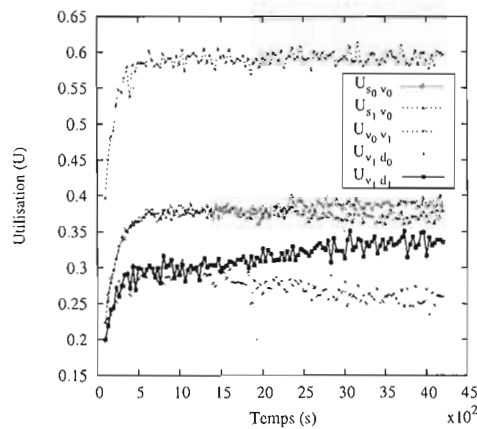


Figure 5.11 Utilisation du lien.

La figure 5.11 montre l'utilisation de chaque lien du réseau simulé. Le lien entre

les deux nœuds de cœur v_0 et v_1 à une utilisation autour de 60 %. Cette valeur constitue la plus importante utilisation par rapport aux autres liens. Elle peut être utilisée pour choisir le taux de perte des rafales de référence afin de maximiser l'utilisation du réseau. En effet, si on vise à réduire le taux de perte, on est obligé de réduire le taux d'injection des rafales dans le réseau et par suite, l'utilisation va diminuer. Ainsi, il faut choisir un taux de perte de référence qui n'est pas très faible pour ne pas diminuer l'utilisation des liens. Le taux de référence ne doit pas aussi être très importante pour pouvoir atteindre la qualité de service requise en termes de taux de perte. Par conséquent, il y a un compromis à établir entre un taux de perte acceptable (pour la qualité de service) et une utilisation acceptable (pour exploiter au maximum les ressources).

5.2.2 Contrôle du taux de perte avec feedback et avec QoS

Dans cette section, les simulations présentées visent à tester l'extensibilité du modèle proposé avec feedback en ajoutant la qualité de service.

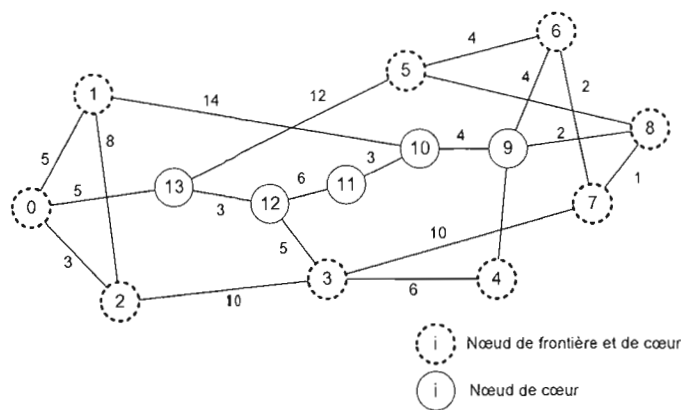


Figure 5.12 Le réseau NSFNet.

La figure 5.12 représente le réseau NSFNet (Zalesky et al., 2007) utilisé pour les simulations. Il est constitué de 14 nœuds et 21 liens bi-directionnels. La figure montre aussi le délai de propagation de chaque lien exprimé en millisecondes. Les nœuds en pointillé sont considérés en même temps comme nœuds de frontière et nœuds de cœur.

Les nœuds tracés en ligne continue sont seulement des nœuds de cœur. Chaque nœud d'entrée s_i génère des rafales qui suivent *une distribution de Poisson* vers chaque nœud de sortie d_j . Il y a trois classes de rafales : c_0 , c_1 et c_2 . La classe c_0 est celle qui a la plus haute priorité. Les débits moyens des rafales pour chaque classe c_0 , c_1 et c_2 sont respectivement 35000, 65000 et 115000 rafales par seconde. Pour la réservation des longueurs d'onde, la préemption est utilisée, c'est-à-dire une rafale de haute priorité peut annuler la réservation d'une rafale moins prioritaire et prendre sa place.

La configuration de la simulation est la suivante :

- La taille maximale d'une rafale 120 KB.
- Le délai de traitement d'un paquet de contrôle est de 10 μ s.
- La valeur de l'offset est égale à 1 ms.
- Les nœuds sont capables d'effectuer la conversion de longueurs d'onde pour les rafales.
- Il n'y a pas de mémoire tampon pour conserver les rafales au niveau des nœuds.
- Pour le routage, l'algorithme du plus court chemin est utilisé.
- Chaque lien contient 70 canaux, chacun opère à 10 Gbps : 15 canaux sont utilisés comme canaux de contrôle et les canaux restants sont utilisés pour les données.
- Le mécanisme de réservation est le *Just-Enough-Time* (JET) (Jue et Vokkarane, 2005).
- Les taux de perte des rafales de référence pour les classes c_0 , c_1 et c_2 sont respectivement fixées à 0, 0,001 et 0,005.
- La technique de la représentation des lieux des pôles donne un gain $K_{s_i d_j c_k}$ égal à 0,3.

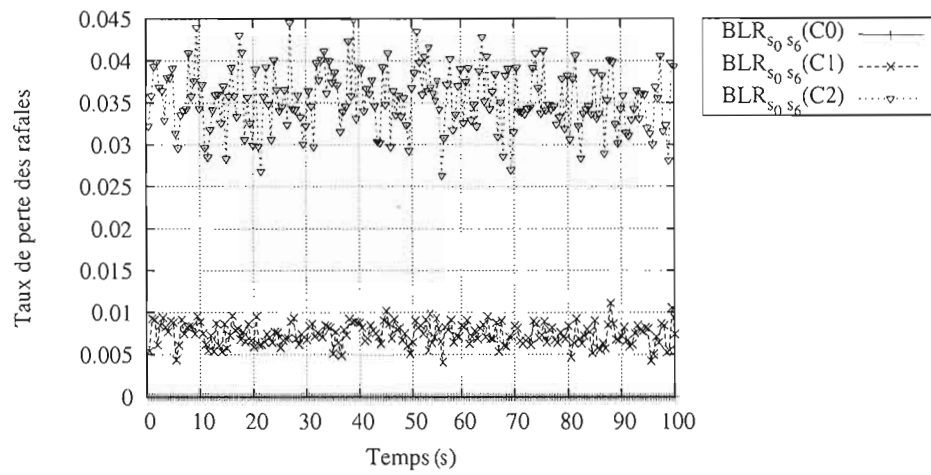


Figure 5.13 Taux de perte des rafales de la source s_0 à la destination d_6 (sans contrôleur).

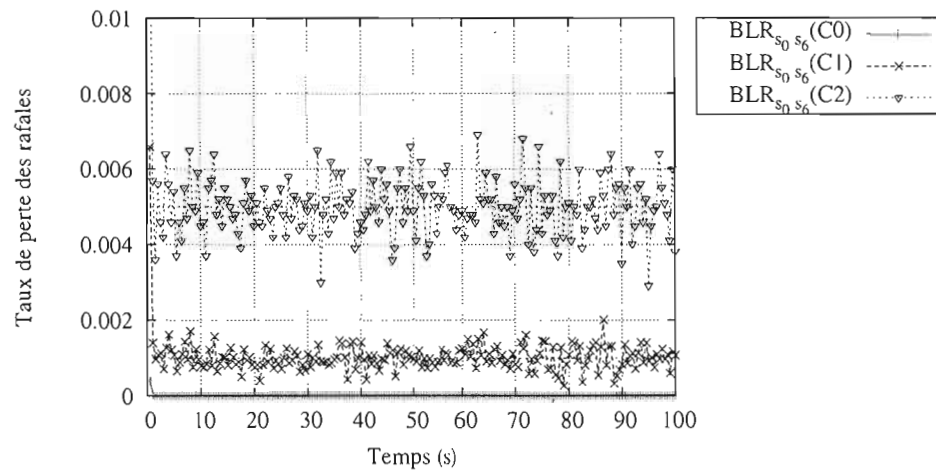


Figure 5.14 Taux de perte des rafales de la source s_0 à la destination d_6 (avec contrôleur).

La figure 5.13 montre le taux de perte des rafales de la source s_0 vers d_6 pour chaque classe de service sans utiliser le contrôleur. Le taux de perte BLR de la classe c_0 est nul puisque la préemption est utilisée. Ainsi, les rafales de la classe c_0 peuvent

prendre la place de n'importe quelle autre rafale de priorité inférieure et de réserver une longueur d'onde. On remarque que le taux de perte des rafales est autour de 0,008 pour la classe c_1 et autour de 0,04 pour la classe c_2 . Dans ce cas, l'inconvénient est que l'administrateur ne peut pas contrôler ce taux de perte.

La figure 5.14 montre le taux de perte des rafales pour les classes c_0 , c_1 et c_2 utilisant le système asservi en boucle fermée. Le taux de perte de rafales pour la classe c_0 reste égal à zéro à cause de la préemption. Cependant, il oscille autour de 0,001 pour la classe c_1 qui est la valeur désirée pour la classe c_1 . Pour la classe c_2 , le taux de perte oscille aussi autour de taux désiré 0,005. L'avantage de l'utilisation du contrôleur est qu'il génère dynamiquement le débit de burstification pour chaque classe. Sans contrôleur, il est impossible d'estimer le débit de burstification de chaque classe qui garantit l'obtention du taux de perte désiré correspondant à celle-ci.

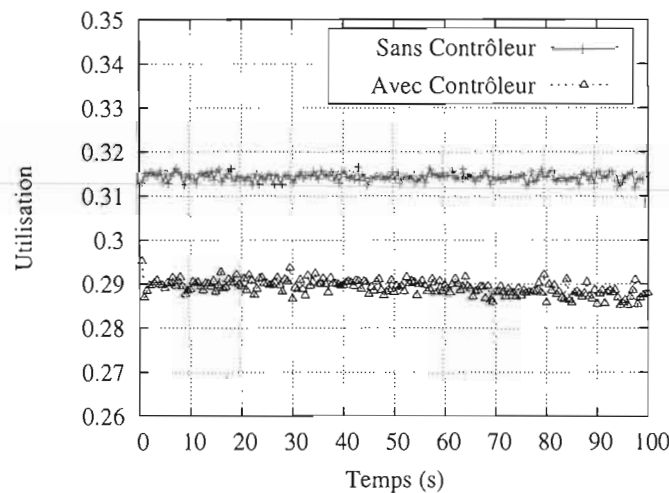


Figure 5.15 Utilisation du réseau.

La figure 5.15 montre l'utilisation du réseau. Sans contrôleur, l'utilisation est autour de 31,5 % contre 29 % avec le contrôleur. Cela est justifié par le fait que le contrôleur fait passer moins de rafales afin de garantir le taux de perte désiré pour chaque classe. Ainsi, il permet d'avoir le maximum d'utilisation des ressources avec le

taux de perte désiré pour chaque classe de service.

5.2.3 Contrôle du taux de perte sans feedback avec prévision du trafic TCP et sans QoS

Les résultats présentés dans ce paragraphe visent à tester le modèle proposé sans feedback et qui utilise la prévision du taux de perte des segments TCP pour estimer le taux de perte des rafales. Ainsi, aucun feedback du réseau n'est nécessaire (fig. 5.4). Nous avons simulé le réseau NSFNet tel que décrit dans la figure 5.12. Les hypothèses de la simulation sont les suivantes :

- Il y a 20000 connexions FTP entre chaque couple de nœuds de frontière.
- Les arrivées des sessions FTP suivent une distribution de Poisson d'inter-arrivée moyenne de 0,01 s.
- Les connexions FTP transmettent des fichiers de tailles différentes suivant la *distribution de Pareto* de moyenne 10 MB et de paramètre 1,2 (Charzinski, 2000).
- TCP Tahoe est utilisé. Cette version de TCP inclue les mécanismes *Slow-Start*, *Congestion Avoidance* et *Fast Retransmit* (Jacobson, 1988; Fall et Floyd, 1996).
- La taille d'un message TCP est de 1500 octets, la taille de l'acquittement TCP (ACK) est de 40 octets.
- La taille maximale de la fenêtre TCP est de 64 KB (≈ 43 segments TCP).
- Le temps de retransmission RTO est estimé à 0,5 s. Ainsi, la taille de l'intervalle I est fixé à 2 s.
- La taille maximale d'une rafale 120 KB (≈ 77 segments).
- Le délai de traitement d'un paquet de contrôle est de 10 μ s (microseconde).
- La valeur de l'offset est égale à 1 ms.
- Les nœuds sont capables d'effectuer la conversion de longueurs d'onde pour les rafales.
- Il n'y a pas de mémoire tampon (buffer) pour conserver les rafales au niveau des nœuds.

- Pour le routage, l'algorithme du plus court chemin est utilisé.
- Chaque lien contient quatre canaux, chacun opère à 10 Gbps : un canal est utilisé pour l'envoi des paquets de contrôle et les canaux restants sont utilisés pour les rafales.
- Le mécanisme de réservation est le *Just-Enough-Time* (JET).
- Il y a une seule classe de service.
- Lorsque le contrôleur est utilisé, le taux de perte des rafales de référence est fixé à 10^{-2} .
- La technique de la représentation des lieux des pôles donne un gain K_{s,d_j} égal à 500.
- Les valeurs réelles (non estimées) du taux de perte des rafales et du taux de perte des segments sont mesurées directement lors de la simulation.
- Les statistiques de la distribution des tailles des rafales montre que plus de 40 % des rafales dans le réseau sont formées de 77 segments. Cette valeur correspond à la taille maximale des rafales. La distribution des tailles des paquets perdus (L_v) montre que plus de 90 % des rafales perdues sont de taille 77 segments. Les rafales de grande taille ont plus de probabilité d'être perdues. En effet, il est plus difficile de trouver des espaces libres dans les canaux pour les grandes rafales.

Les simulations visent à évaluer la performance de la prévision du taux de perte des segments (c'est-à-dire le taux de perte des segments (SLR) comparé au taux estimé de perte des segments (ESLR)). Ensuite, le taux de perte des rafales (BLR) est comparé au taux estimé de perte des rafales (EBLR). Enfin, nous comparons le contrôleur avec feedback et le contrôleur sans feedback qui utilise seulement le taux de perte estimé par la prévision TCP au niveau du nœud d'entrée.

La figure 5.16 montre le taux de perte réel (mesuré) des segments (SLR) comparé au taux de perte estimé des segments (ESLR) entre la source s_0 et la destination d_6 . La figure montre que l'erreur entre ESLR et SLR est faible ($RMSE = 0,00344$).

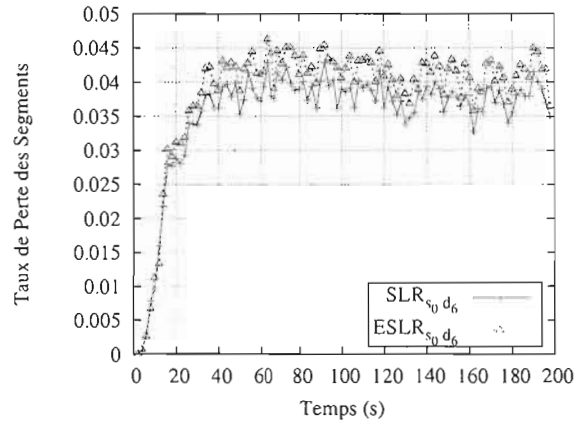


Figure 5.16 Taux de perte des segments réel (mesuré) et estimé entre la source s_0 et la destination d_6 (sans contrôleur).

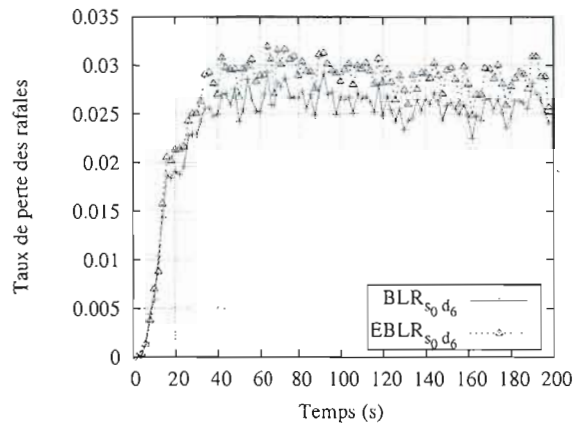


Figure 5.17 Taux de perte des rafales réel (mesuré) et estimé entre la source s_0 et la destination d_6 (sans contrôleur).

La figure 5.17 montre le taux de perte des rafales mesuré (BLR) entre la source s_0 à la destination d_6 comparé au taux estimé (EBLR) déduit à partir du ESLR à l'aide de l'équation 5.28. Il est clair que l'erreur entre la valeur réelle et la valeur estimée reste faible ($RMSE = 0,002868$).

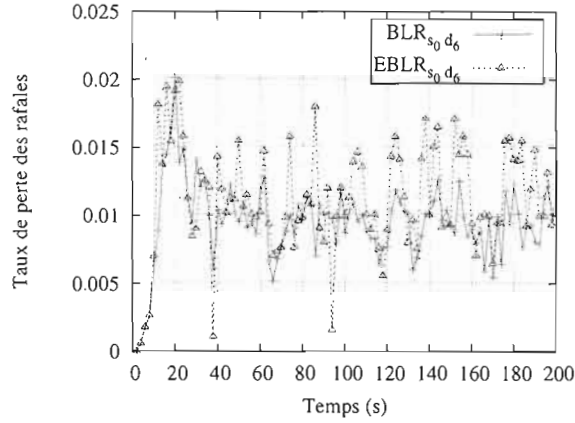


Figure 5.18 Taux de perte des rafales (BLR) contrôlé avec EBLR entre la source s_0 et la destination d_0 .

La figure 5.18 montre le taux de perte des rafales (BLR) obtenu en utilisant le contrôleur sans feedback (fig. 5.4). Le taux de perte estimé $EBLR$ a été utilisé à la place du taux de perte réel BLR comme entrée au contrôleur pour adapter le débit de burstification. La figure montre que l'erreur entre le taux de perte des rafales estimée $EBLR$ et le taux de perte mesuré BLR n'est pas importante et que l'objectif de stabiliser le taux de perte BLR est atteint. Ainsi, le taux de perte des rafales (BLR) oscille autour du taux de perte de référence fixé par l'administrateur (0,01). Cela montre que le BLR estimé peut remplacer la valeur mesurée du BLR. Ainsi, le feedback du réseau n'est plus nécessaire. Le nœud d'entrée devient autonome et capable de contrôler le taux de perte de rafales à l'intérieur du réseau sans aucun feedback du réseau.

5.3 Conclusion

Dans ce chapitre, nous avons utilisé la modélisation et la prévision du trafic combinées avec les modèles de systèmes asservis avec et sans feedback afin de contrôler le taux de perte des rafales dans un réseau OBS. Le modèle proposé permet de garantir un taux de perte pour le réseau qui est fixé par l'administrateur du réseau. Ensuite, nous avons étendu le modèle pour tenir compte de la qualité de service en définissant des

classes de service. Chaque classe de service est définie par une valeur de taux de perte. Nous avons effectué des simulations pour le modèle proposé sur une topologie simple et une topologie complexe (NSFNet). Les résultats montrent que la technique proposée permet de contrôler le taux de perte pour chaque classe de service. Ainsi, les valeurs désirées pour le taux de perte sont atteintes et préservées par le contrôleur. Aucune garantie ne serait possible sans l'approche proposée. Bien que l'utilisation du réseau diminue lorsqu'on utilise la technique proposée, elle permet de garantir le taux de perte pour chaque classe de service.

Le modèle de contrôle proposé a été ensuite amélioré afin d'éviter les feedbacks à partir du réseau. Ainsi, le nœud d'entrée prévoit le taux de perte sans aucune information provenant du réseau. Nous avons proposé une méthode pour estimer le taux de perte des rafales en se basant sur les statistiques TCP disponibles au niveau du nœud d'entrée. Ainsi, le taux de perte des segments TCP est estimé en utilisant les statistiques sur les retransmissions TCP. Ensuite, le taux de perte des rafales (niveau OBS) est calculé à partir du taux de perte des segments TCP. Les résultats montrent que les taux de perte des rafales estimés sont précis.

Nous avons ensuite intégré cette méthode d'estimation dans le modèle de contrôle proposé afin d'éviter le feedback du réseau. Les simulations de la topologie NSFNet montrent que le modèle amélioré contrôle efficacement le taux de perte sans aucun feedback provenant du réseau ni de mesure de bout en bout.

D'autres études peuvent être considérées pour évaluer la précision de la prévision du taux de perte des rafales à partir des statistiques au niveau TCP, mais en utilisant d'autres variantes de TCP telles que HighSpeed TCP (Floyd, 2003) et FAST (Paganini et al., 2003) qui sont proposées spécialement pour les réseaux haut débit. En effet, ces mécanismes changent le mécanisme de contrôle de congestion du protocole TCP pour l'adapter aux réseaux haut-débit et par suite, le trafic généré changera de caractéristiques et les estimations des pertes vont être différentes.

CONCLUSION

Le travail présenté dans cette thèse a permis, dans un premier temps, d'identifier et d'étudier les différents paramètres et techniques permettant d'améliorer la prévision du trafic. Ainsi, les résultats obtenus ont permis de déduire un certain nombre de recommandations pour paramétrer les modèles de prévision. Une technique d'échantillonnage adaptée à la prévision du trafic a été aussi proposée. Dans un second temps, la prévision a été exploitée pour la gestion des files d'attente dans les routeurs et le contrôle du taux de pertes dans les réseaux optiques à commutation de rafales.

Nous avons tout d'abord présenté les caractéristiques du trafic Internet, les principaux modèles de prévision du trafic, les prétraitements possibles sur les données ainsi que les travaux récents dans le domaine de la modélisation et la prévision du trafic.

Afin d'étudier l'effet des différents paramètres sur la précision de la prévision, nous avons utilisé deux ensembles de traces de trafic réel collectées sur deux liens de capacités différentes (100 Mbps et 1 Gbps). Plusieurs prévisions ont été effectuées en utilisant les modèles ARMA, ARIMA et α -SNF. Nous avons étudié le choix des entrées du modèle de prévision. Les résultats montrent qu'un seul retard est suffisant pour la prévision du débit. Nous avons aussi trouvé que la prévision se dégrade avec l'augmentation de la variance ou de la granularité. De plus, l'augmentation de la quantité de données pour l'apprentissage n'améliore nécessairement pas la performance du modèle de prévision. Nous avons aussi trouvé que le comportement du trafic dépend principalement des grands paquets (de tailles supérieure 800 octets). Cela nous a incité à étudier l'utilisation des variables exogènes comme entrées du modèle. Ainsi, le nombre de paquets de grande taille, utilisée comme entrée du modèle de prévision, améliorent la précision de la prévision du débit.

Les techniques de prétraitement, notamment celles d'échantillonnage, ont été aussi étudiées dans ce travail. Nous avons montré que l'application des techniques d'échantillonnage a un effet considérable sur les caractéristiques des données obtenues et sur la performance de la prévision. Ainsi, nous avons proposé une technique d'échantillonnage qu'on a baptisée *échantillonnage basé sur le maximum* (*maximum-based sampling*). Cette technique permet d'utiliser l'historique des pics de débit pour prévoir les pics qui présentent les éventuelles futures congestions. Nous avons prouvé, analytiquement et par simulations, que cette technique a l'avantage, par rapport aux autres techniques, de préserver l'auto-similarité et la dépendance à long terme. Les simulations montrent aussi que l'échantillonnage basé sur le maximum améliore la performance de la prévision en termes d'erreur.

Ensuite, nous avons proposé d'utiliser la prévision dans le cadre de la gestion des files d'attente. Nous avons proposé un nouveau mécanisme, baptisé α -SNFAQM, qui a la particularité de prévoir le débit du trafic et le type des futures congestions : congestion grave et congestion légère. Le mécanisme α -SNFAQM vise à stabiliser la taille de la file d'attente tout en maximisant l'utilisation du lien. Une étude analytique a été développée pour justifier les décisions du mécanisme α -SNFAQM et estimer la taille moyenne de la file (en considérant seulement le trafic TCP et en supposant que les temps d'aller-retour et les taux de perte sont identiques pour toutes les connexions TCP). Le mécanisme a été aussi implémenté et testé avec des différentes topologies et configurations. La performance du mécanisme α -SNFAQM dépasse celle des autres mécanismes tels que RED, PAQM ou APACE. En effet, les résultats montrent que le α -SNFAQM permet de stabiliser la taille de la file d'attente tout en gardant une utilisation importante du lien. Cela est important pour contrôler les temps d'attente dans les routeurs et par suite, contrôler le délai de bout-en-bout dans le réseau. Le mécanisme α -SNFAQM permet aussi de maximiser le débit à la réception (au niveau de la couche application) en minimisant les retransmissions TCP.

Une autre application de la modélisation et la prévision du trafic a été proposée dans le cadre des réseaux optiques à commutation de rafales. En effet, la prévision a été

utilisée en combinaison avec les modèles de systèmes asservis avec et sans feedback afin de contrôler le taux de perte des rafales pour chaque classe de service.

Les résultats des simulations montrent que le modèle de contrôle proposé permet de contrôler et stabiliser le taux de perte pour chaque classe de service autour des valeurs désirées précisées par l'administrateur.

Le modèle de contrôle a été ensuite amélioré pour éviter les feedbacks à partir du réseau. L'amélioration consiste à prévoir le taux de perte des rafales en fonction du taux de perte des segments TCP ; celui-ci étant estimé en utilisant les statistiques sur les retransmissions TCP. Les résultats montrent que le modèle amélioré contrôle efficacement le taux de perte sans aucun feedback du réseau ni de mesure de bout en bout.

Plusieurs perspectives de recherche sont possibles pour ce travail. Il est intéressant d'effectuer une analyse de l'effet des autres paramètres du réseau, tels que le nombre de flux et leur nature, sur les modèles de prévision. Nous pouvons aussi envisager des prévisions sur des variables, autres que le débit, telles que le délai, le taux de perte ou le nombre de flux. Cela permet de prévoir la performance du réseau et par suite, servira à adapter les paramètres des protocoles réseaux à son état futur et garantir une meilleure qualité de service. Des prévisions du trafic par protocole (UDP, TCP) ou par application (Web, Peer to Peer, FTP) peuvent être aussi effectuées afin de prévoir l'évolution du trafic pour chaque type de protocole ou d'application.

D'autre part, le mécanisme α -SNFAQM peut être amélioré pour tenir compte de différentes classes de service. Ainsi, la prévision du débit de chaque classe de trafic peut servir pour décider des rejets des paquets.

La technique de l'échantillonnage basé sur le maximum, combinée avec un modèle de prévision, peut être exploitée afin d'effectuer la réservation dynamique de la bande passante disponible pour les réseaux garantissant la qualité de service.

Des prévisions à long terme peuvent être envisagées au niveau de chaque lien d'un

réseau fédérateur. Ainsi, il sera possible d'avoir une idée plus claire sur les éventuels changements et améliorations nécessaires à l'avenir pour les équipements et les liens du réseau.

APPENDICE A

RAPPEL SUR LES PROCESSUS ALÉATOIRES

Définition A.1. Variable aléatoire : soit (Ω, \mathcal{F}, P) un espace de probabilité. Une application X de Ω dans \mathbb{R} est appelée une variable aléatoire. L'ensemble des valeurs prises par la variable aléatoire X est l'image de Ω par X et est noté $X(\Omega) \subset \mathbb{R}$.

X est une variable aléatoire réelle discrète si $X(\Omega)$ est dénombrable.

Définition A.2. Processus stochastique : un processus stochastique (ou aléatoire) est une famille de variables aléatoires (c'est-à-dire, des applications mesurables) définies sur le même espace de probabilité (Ω, \mathcal{F}, P) indexé par T et à valeurs dans \mathbb{R} . Un processus stochastique est noté par $\{X(t); t \in T\}$. La valeur de la variable aléatoire $X(t)$ en un certain $\omega \in \Omega$ est désignée par $X(t)(\omega)$.

Définition A.3. Soit X une variable aléatoire discrète.

L'espérance (ou moyenne) de X est définie par $\mathbb{E}[X] = \sum_k kP[X = k] = \mu_X$

et on a $\mathbb{E}[X^2] = \sum_k k^2P[X = k]$.

La variance de X est définie par $\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \sigma_X^2$.

La covariance de deux variables aléatoires X et Y est définie par :

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y].$$

Le coefficient de corrélation entre X et Y est définie par :

$$r(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}.$$

Définition A.4. Loi uniforme : une variable aléatoire X suit une loi de probabilité uniforme sur un intervalle $[a, b]$ si elle est associée à une densité de probabilité égale à

$$p(x) = \begin{cases} \frac{1}{b-a} & \text{pour } a \leq x \leq b, \\ 0 & \text{sinon.} \end{cases}$$

Définition A.5. Égalité en distribution : deux variables aléatoires X et Y sont égales en distribution (on note $X =_d Y$) si elles ont la même distribution, c.-à-d. :

$$P(X \leq x) = P(Y \leq x) \quad \forall x \in \mathbb{R}.$$

Définition A.6. Fonction de distribution conjointe : soient $Y(t_1), \dots, Y(t_n)$ des variables aléatoires continues. La fonction de distribution conjointe de $Y(t_1), \dots, Y(t_n)$, notée $f_{Y(t_1), \dots, Y(t_n)}$, est définie par :

$$\begin{aligned} f_{Y(t_1), \dots, Y(t_n)} : \mathbb{R}^n & \rightarrow \mathbb{R} \\ (x_1, x_2, \dots, x_n) & \rightarrow f_{Y(t_1), \dots, Y(t_n)}(x_1, x_2, \dots, x_n) \\ & = P[Y(t_1) \leq x_1, Y(t_2) \leq x_2, \dots, Y(t_n) \leq x_n]. \end{aligned}$$

Définition A.7. Coefficient d'autocorrélation d'un processus :

soit $X = \{X(t); t \geq 0\}$ un processus aléatoire.

L'autocovariance du processus X entre les instants s et t est définie par :

$$\text{Cov}(X(s), X(t)) = \mathbb{E}[(X(t) - \mu_{X(t)})(X(s) - \mu_{X(s)})]$$

où $\mu_{X(s)}$ et $\mu_{X(t)}$ sont les espérances respectives de $X(s)$ et $X(t)$.

Le coefficient d'autocorrélation du processus X entre les instants s et t est défini par :

$$r(X(s), X(t)) = \frac{\text{Cov}(X(s), X(t))}{\sigma_{X(t)} \sigma_{X(s)}} = \frac{\mathbb{E}[(X(t) - \mu_{X(t)})(X(s) - \mu_{X(s)})]}{\sigma_{X(t)} \sigma_{X(s)}}.$$

Si le processus X est stationnaire, alors $\mathbb{E}[X(t)] = \mu$ et $\text{Var}(X(t)) = \sigma$ sont constantes et le coefficient d'autocorrélation du processus ne dépend que du décalage (noté k) entre deux instants. Ainsi, le coefficient d'autocorrélation s'écrit :

$$r(k) = \frac{\mathbb{E}[(X(t) - \mu)(X(t+k) - \mu)]}{\sigma^2}.$$

Définition A.8. Bruit blanc (white noise) : un processus $\{\epsilon(t); t \geq 0\}$ est un bruit blanc si les variables aléatoires $\epsilon(t)$ sont non corrélées, d'espérance nulle et de variance constante. Autrement dit, pour tout t

$$\mathbb{E}[\epsilon(t)] = 0, \quad \text{Var}[\epsilon(t)] = \sigma^2, \quad \text{Cov}(\epsilon(t), \epsilon(t+h)) = 0 \quad \text{pour } h \neq 0.$$

D'après la définition, il est simple de déduire qu'un bruit blanc est stationnaire.

Définition A.9. Loi de Gauss : une variable aléatoire réelle X suit une loi de Gauss d'espérance μ et d'écart type σ strictement positif (donc de variance σ^2) si elle admet pour densité de probabilité la fonction $p(x)$ définie, pour tout nombre réel x , par :

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Définition A.10. Bruit blanc Gaussien : un processus $\{\epsilon(t); t \geq 0\}$ est un bruit blanc Gaussien s'il est un bruit blanc dont la distribution de chaque variable aléatoire $\epsilon(t)$ est Gaussienne.

Définition A.11. Un processus $X(t)$ tel que $X(0) = 0$ est à accroissements indépendants si, pour toute suite finie $0 < t_1 < t_2 \dots < t_n$, les variables aléatoires

$$X(t_1), X(t_2) - X(t_1), \dots, X(t_n) - X(t_{n-1})$$

sont indépendantes.

Définition A.12. Un processus à accroissements indépendants est à accroissements stationnaires si la loi de l'accroissement $(X(t+s) - X(t))$ ne dépend pas de t , pour tout $t \geq 0$.

Définition A.13. Processus de comptage : un processus aléatoire $\{N(t); t \geq 0\}$ à valeurs entières est un processus de comptage si

- i) $N(0) = 0$;
- ii) $\forall s \leq t, N(s) \leq N(t)$.

Définition A.14. Processus de Poisson : un processus aléatoire $\{N(t); t \geq 0\}$ à valeurs entières est un processus de Poisson de paramètre $\lambda > 0$ si

- i) $\{N(t)\}$ est un processus de comptage à accroissements indépendants et stationnaires;
- ii) la variable $N(t)$ suit la loi de Poisson de paramètre λ

$$\forall n \geq 0, P(N(t) = n) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

Définition A.15. On définit l'opérateur « retard » (backward shift operator) noté L par :

$$\left\{ \begin{array}{l} L^i y(t) = y(t - i), \quad \forall i \in \mathbb{N} \\ (1 - L)y(t) = y(t) - y(t - 1), \\ (1 - L)^d y(t) = (1 - L)^{d-1} (1 - L)y(t) \quad \text{si } d \in \mathbb{N}, \\ (1 - L)^d y(t) = \left(\sum_{k=0}^{\infty} \frac{\prod_{a=0}^{k-1} (d-a)}{k!} (-L)^k \right) y(t) = \left(1 - dL + \frac{d(d-1)}{2} L^2 - \dots \right) y(t) \quad \text{si } d \in \mathbb{R}. \end{array} \right.$$

APPENDICE B

LE MÉCANISME *RANDOM EARLY DETECTION* (RED)

Le mécanisme *Random Early Detection* (RED) est un algorithme de gestion pour les files d'attente qui a été proposé par (Floyd et Jacobson, 1993).

Les variables de l'algorithme sont :

- avg : la taille moyenne de la file d'attente exprimée en nombre de paquets,
- th_{min} : le seuil minimal pour la taille moyenne de la file d'attente exprimé en nombre de paquets,
- th_{max} : le seuil maximal pour la taille moyenne de la file d'attente exprimé en nombre de paquets,
- q : la taille actuelle de la file d'attente exprimée en nombre de paquets,
- w_q : un paramètre fixe entre zéro et un,
- p_a : la probabilité de rejet des paquets,
- $maxp$: la probabilité maximale de rejet.

L'algorithme RED est donné par :

Algorithme 1 RED

```

1:  $avg \leftarrow 0$ 
2:  $count \leftarrow -1$ 
3: à l'arrivée de chaque paquet :
4:  $avg \leftarrow (1 - w_q)avg + w_q q$ 
5: si  $th_{min} \leq avg < th_{max}$  alors
6:   incrémenter  $count$ 
7:    $p_b \leftarrow maxp(avg - th_{min}) / (th_{max} - th_{min})$ 
8:    $p_a \leftarrow p_b / (1 - count/p_b)$ 
9:   rejeter le paquet arrivé avec une probabilité  $p_a$ 
10:   $count \leftarrow 0$ 
11: sinon si  $avg \geq th_{max}$  alors
12:   rejeter le paquet
13:    $count \leftarrow 0$ 
14: sinon
15:   accepter le paquet dans la file d'attente
16:    $count \leftarrow -1$ 
17: fin si

```

APPENDICE C

FONCTION DE TRANSFERT D'UN SYSTÈME ASSERVI

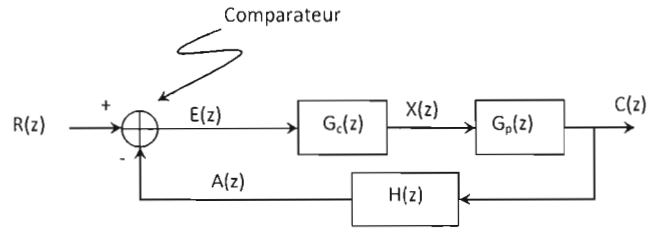


Figure C.1 Système asservi en boucle fermée.

Une fonction de transfert décrit comment l'entrée est transformée en une sortie (Ogata, 1997). Elle est calculée en divisant la transformée en Z de la sortie $C(z)$ par la transformée en Z de l'entrée $R(z)$. Ainsi, elle est donnée par (fig. C.1)

$$T(z) = \frac{C(z)}{R(z)}. \quad (\text{C.1})$$

Les équations suivantes décrivent les fonctions de transfert du système asservi en boucle fermée (fig. C.1) :

$$C(z) = X(z) \times G_p(z), \quad (\text{C.2})$$

$$X(z) = E(z) \times G_c(z). \quad (\text{C.3})$$

Par suite, on a :

$$C(z) = E(z) \times G_c(z) \times G_p(z). \quad (\text{C.4})$$

On a aussi :

$$A(z) = C(z) \times H(z), \quad (\text{C.5})$$

$$E(z) = R(z) - A(z), \quad (\text{C.6})$$

donc

$$R(z) = E(z) + C(z) \times H(z). \quad (\text{C.7})$$

D'après l'équation C.4, la fonction de transfert du système en boucle ouverte est :

$$W(z) = \frac{C(z)}{E(z)} = G_c(z) \times G_p(z). \quad (\text{C.8})$$

D'après les équations C.7 et C.4, la fonction de transfert du système en boucle fermée est :

$$T(z) = \frac{C(z)}{R(z)} = \frac{G_c(z) \times G_p(z)}{1 + G_c(z) \times G_p(z) \times H(z)}. \quad (\text{C.9})$$

APPENDICE D

STABILITÉ ET REPRÉSENTATION DES LIEUX DES PÔLES

– Stabilité du système :

Un système asservi est stable si la valeur de la variable à contrôler reste bornée au cours du temps, c.-à-d. elle doit éventuellement osciller autour d'un point de référence. Ainsi, le système est instable si la variable à contrôler est non bornée, cela implique qu'elle ne peut jamais atteindre la valeur de référence désirée (Ogata, 1997).

Une fonction de transfert décrit comment l'entrée est transformée en une sortie (Ogata, 1997). Elle est calculée en divisant la transformée en Z de la sortie par la transformée en Z de l'entrée. Les racines du numérateur de la fonction de transfert sont appelées *les zéros*. Les racines du dénominateur de la fonction de transfert sont appelées *les pôles*.

La valeur des pôles de la fonction de transfert est déterminante pour son comportement dynamique et sa stabilité (Ogata, 1997; Hellerstein et al., 2004; Parekh et al., 2001). Si l'un des pôles a un module ($|z|$) supérieur à 1, alors la fonction de transfert n'est pas stable (Hellerstein et al., 2004; Parekh et al., 2001). Rappelons que z est un nombre complexe ainsi les pôles et les zéros sont des nombres complexes qu'on peut représenter dans le plan et qui sont caractérisés par des longueurs (modules). Ainsi, un système représenté par sa fonction de transfert est stable si tous les pôles sont à l'intérieur du cercle de rayon 1.

Par exemple, la fonction de transfert donnée par l'équation D.1 a $(0, 2)$ comme zéro et $(0, 3)$ comme pôle.

$$W(z) = \frac{z - 0,2}{z - 0,3}. \quad (\text{D.1})$$

La fonction de transfert $W(z)$ est considérée stable puisque le module de son pôle est égal à $(0, 3)$.

Dans ce qui suit, nous donnons un exemple de système asservi et nous détaillons davantage la technique de la représentation des lieux des pôles.

– Exemple de système asservi :

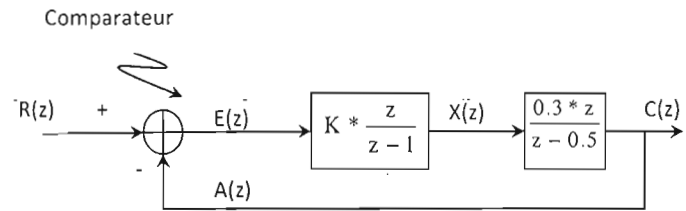


Figure D.1 Exemple d'un système asservi en boucle fermée.

La figure D.1 montre un exemple de système asservi en boucle fermée. En appliquant les formules détaillées à l'appendice C, les fonctions de transfert pour cet exemple sont comme suit.

La fonction de transfert du système en boucle ouverte est (d'après l'équation C.8) :

$$W(z) = \frac{C(z)}{E(z)} = \frac{K \times z}{z - 1} \times \frac{0,3 \times z}{z - 0,5}. \quad (\text{D.2})$$

La fonction de transfert du système en boucle fermée est (d'après l'équation C.9) :

$$T(z) = \frac{C(z)}{R(z)} = \frac{(K \times z)(0,3 \times z)}{(z-1) \times (z-0,5) + K \times z \times (0,3 \times z)}. \quad (\text{D.3})$$

On définit l'équation caractéristique comme étant le dénominateur de la fonction de transfert à boucle fermée égalisé à 0 (éq. D.4). Les pôles de la boucle fermée sont les racines de l'équation caractéristique.

$$(z-1) \times (z-0,5) + K \times z \times (0,3 \times z) = 0. \quad (\text{D.4})$$

Ainsi, on a :

$$K = \frac{-(z-1) \times (z-0,5)}{z \times (0,3 \times z)}. \quad (\text{D.5})$$

La technique de la représentation des lieux des pôles (*root locus*) est ensuite appliquée pour calculer le gain K .

- Représentation des lieux des pôles (*Root Locus*) :

Cette technique a été proposée dans (Evans, 1948; Evans, 1950). Une représentation des lieux des pôles est une représentation de l'évolution de l'ensemble des pôles de la fonction de transfert (c.-à-d. les racines de la fonction caractéristique) lorsque le gain K varie de zéro vers l'infinie. Ceci est représenté graphiquement sur le plan complexe (fig. D.2). L'axe des abscisses correspond à la partie réelle de z ($\text{Réal}(z)$) et l'axe des ordonnées correspond à la partie imaginaire de z ($\text{Im}(z)$). Les deux courbes (verte et bleue) représentent les pôles de la fonction de transfert.

Les lieux des pôles déterminent la stabilité du système. Ainsi, l'idée est de choisir le gain qui correspond aux pôles qui fournissent un système stable. Le pôle doit satisfaire à deux conditions : être à l'intérieur du cercle de rayon 1 pour éviter l'instabilité du système et être sur l'axe des abscisses pour éviter les oscillations (Parekh et al., 2002).

D'après la figure, il y a plusieurs racines possibles pour l'équation caractéristique qui satisfont ces deux conditions. Dans notre exemple, nous choisissons la racine de

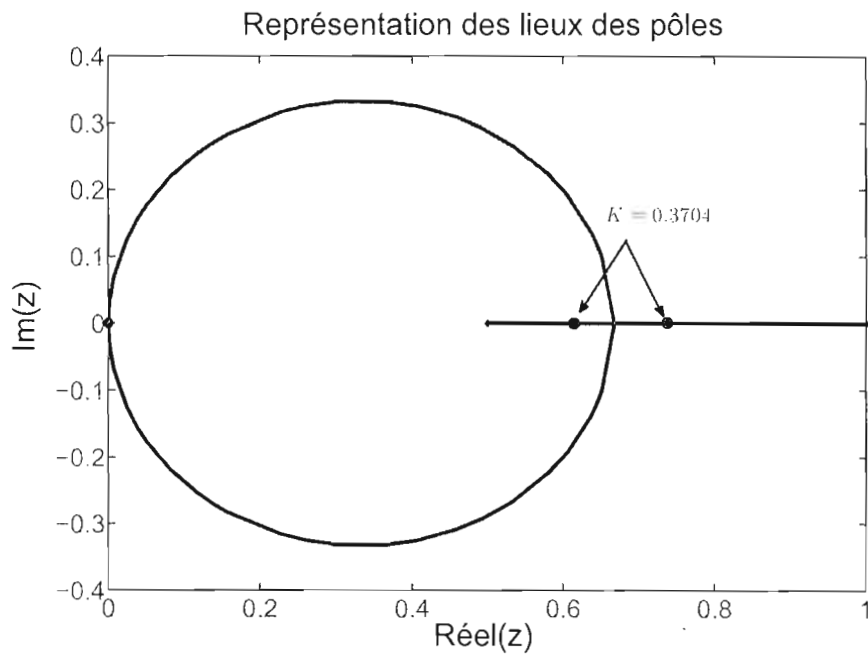


Figure D.2 Exemple de représentation des lieux des pôles.

valeur 0,6. Ainsi, le gain K qui donne cette racine doit être déterminé. Il suffit de remplacer z par 0,6 dans l'équation D.5 pour obtenir le gain K . Ainsi, K est égal à 0,3704. En remplaçant K par sa valeur dans l'équation caractéristique D.4, nous obtenons :

$$(1 + 0,3 \times 0,3704) \times z^2 - 1,5 \times z + 0,5 = 0. \quad (\text{D.6})$$

On peut facilement vérifier que les racines de l'équation D.6 sont (0,6) et (0,75) (fig. D.2). Les deux racines sont de modules inférieurs à 1 ; par conséquent, le système est stable pour le gain choisi ($K = 0,3704$).

Des outils comme *Matlab* peuvent être utilisés pour appliquer cette technique.

BIBLIOGRAPHIE

- Abry, P. et D. Veitch. 1998. « Wavelet analysis of long-range-dependent traffic », *IEEE Trans. Information Theory*, vol. Vol.44, p. 2-15.
- Aldrich, J. 1997. « R. A. Fisher and the Making of Maximum Likelihood 1912-1922 », *Statistical Science*, vol. 12, no. 3, p. 162-176.
- Allman, M., W. M. Eddy, et S. Ostermann. 2003. « Estimating loss rates with TCP », *ACM Performance Evaluation Review*, vol. 31, p. 2003.
- Allman, M., V. Paxson, et W. Stevens. 1999. RFC 2581 : TCP congestion control.
- Aly, W. H. F., M. F. Zhani, et H. Elbiaze. 2007a. « Feedback control for OBS networks », *In proceedings of International Conference on Computer Theory and Applications (ICCTA'07), Alexandria, Egypt*.
- . 2007b. « Using closed loop feedback control theoretic techniques to improve OBS networks performance », *In proceedings of International Workshop on Optical Burst/Packet Switching (WOBS)*.
- . 2008. « On controlling burst loss ratio inside an OBS network », *IEEE Symposium on Computers and Communications, ISCC*, p. 56-61.
- . 2009. « Adaptive offset for OBS networks using feedback control techniques », *In Proceedings of IEEE Symposium on Computers and Communications, ISCC*.
- Anjum, F. et L. Tassiulas. 1999. « Fair bandwidth sharing among adaptive and non-adaptive flows in the Internet », *In Proceedings of IEEE INFOCOM'99, New York, USA*.
- Aoul, Y. H., A. Mehaoua, et C. Skianis. 2007. « A fuzzy logic-based aqm for real-time traffic over Internet », *Computer Networks*, vol. 51, no. 16, p. 4617 - 4633.
- Athuraliya, S., V. H. Li, S. H. Low, et Q. Yin. 2001. « REM : Active queue management », *IEEE Network*, p. 48-53.
- Baldine, I., H. G. Perros, G. N. Rouskas, et D. Stevenson. 2002. « Jumpstart : A just-in-time signaling architecture for WDM burst-switched networks », *IEEE Communications Magazine*, vol. 40, p. 82-89.

- Barakat, C., P. Thiran, G. Iannaccone, C. Diot, et P. Owezarski. 2003. « Modeling Internet backbone traffic at the flow level », *IEEE Transactions on Signal Processing*, vol. 51, no. 8.
- Bermolen, P. et D. Rossi. 2009. « Support vector regression for link load prediction », *Computer Networks*, vol. 53, no. 2, p. 191 – 201. QoS Aspects in Next-Generation Networks.
- Box, G. E. P. et G. M. Jenkins. 1976. *Time Series Analysis, Forecasting, and Control*. San Francisco : Holden-Day.
- Braden, B., D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, et L. Zhang. 1998. « RFC 2309 : Recommendations on queue management and congestion avoidance in the Internet ».
- Brakmo, L. S. et L. L. Peterson. 1995. « TCP Vegas : End to end congestion avoidance on global Internet », In *IEEE Journal of Selected Areas in Communications*, vol. 13, p. 1465–1480.
- Bush, S. F. 1999. « Active virtual network management prediction », In *Proceedings of the 13th Workshop on Parallel and Distributed simulation*, p. 182–192.
- C. Qiao, M. Y. 2000. « Choices features and issues in optical burst switching », *Optical Networks Magazine*, vol. 2, no. 1, p. 36–44.
- Cantieni, G., G. Iannaccone, C. Barakat, C. Diot, et P. Thiran. 2006. « Reformulating the monitor placement problem : Optimal network-wide sampling ». In *40th Annual Conference on Information Sciences and Systems*, p. 1725 –1731.
- Casetti, C., J. F. Kurose, et D. F. Towsley. 1996. « A new algorithm for measurement-based admission control in integrated service packet networks », In *Proceedings of the International Workshop on Protocols for High-Speed Networks*, p. 13–28.
- Casolari, S. et M. Colajanni. 2009. « Short-term prediction models for server management in Internet-based contexts », *Decision Support Systems*, vol. 48, no. 1, p. 212 – 223. Information product markets.
- Chabchoub, Y., C. Fricker, F. Guillemin, et P. Robert. 2010. « On the statistical characterization of flows in Internet traffic with application to sampling », *Computer Communications*, vol. 33, no. 1, p. 103 – 112.
- Charzinski, J. 2000. « HTTP/TCP connection and flow characteristics », *Performance Evaluation*, vol. 42, p. 149–162.
- Choi, B. 1992. *ARMA Model Identification*. New York : Springer-Verlag.
- Cohen, J. 1988. « Statistical power analysis for the behavioral sciences (2nd ed.) », *Hillsdale, NJ : Lawrence Erlbaum Associates*.

- Cozzani, I. et S. Giordano. 1998. « Traffic sampling methods for end-to-end qos evaluation in large heterogeneous networks », *Computer Networks and ISDN Systems*, vol. 30, no. 16-18, p. 1697 – 1706.
- Crovella, M. E. et A. Bestavros. 1997. « Self-similarity in World Wide Web traffic : Evidence and possible causes », *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, p. 835–846. Revised and substantially corrected version of [Crovella and Bestavros, 1996].
- Dempster, A., N. Laird, et D. Rubin. 1977. « Maximum likelihood for incomplete data via the em algorithm », *Journal of the Royal Statistical Society*, vol. 39, p. 1–38.
- Detti, A., V. Eramo, et M. Listanti. 2002. « Optical burst switching with burst drop (pbs/bd) : An easy improvement », *IEEE International Conference on Communications*, vol. 5, p. 2687–2691.
- Doulamis, A., N. Doulamis, et S. Kollias. 2000. « Traffic prediction and network resources estimation of VBR MPEG-2 sources using adaptively trained neural networks », *10th Mediterranean Electrotechnical Conference, MELECON*, vol. 2, p. 717–720.
- Downey, A. B. 2001. « Evidence for long-tailed distributions in the Internet ». In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, p. 229–241. ACM Press.
- Duffield, N., C. Lund, et M. Thorup. 2005. « Estimating flow distributions from sampled flow statistics », *IEEE/ACM Trans. Netw.*, vol. 13, no. 5, p. 933–946.
- Duffield, N. G. et M. Grossglauser. 2000. « Trajectory sampling for direct traffic observation », *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 4, p. 271–282.
- Elbiaze, H., M. F. Zhani, O. Cherkaoui, et F. Kamoun. 2009. « A new structure-preserving sampling method for predicting self-similar traffic », *Telecommunication Systems Journal- Modeling, Analysis, Design and Management*, vol. 43, no. 3-4.
- Estan, C. et G. Varghese. 2002. « New directions in traffic measurement and accounting », *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, p. 323–336.
- Evans, W. R. 1948. « Graphical analysis of control systems », *Transactions AIEE*, vol. 67, p. 547–551.
- . 1950. « Control systems synthesis by root locus method », *Transactions AIEE*, vol. 69, p. 66–69.
- Fall, K. et S. Floyd. 1996. « Simulation-based comparisons of Tahoe, Reno, and Sack TCP », *Computer Communication Review*, vol. 26, p. 5–21.
- Farahmand, F., Q. Zhang, et J. Jue. 2004. « A feedback-based contention avoidance

- mechanism for optical burst switching networks », *Proceedings, 3rd International Workshop on Optical Burst Switching*.
- Farahmand, F., Q. Zhang, et J. P. Jue. 2007. « A feedback-based congestion control mechanism for labeled optical burst switched networks », *Photonic Network Communications*, p. 307–316.
- Farraposo, S., P. Owezarski, et E. Monteiro. 2006. « Contribution of anomalies detection and analysis on traffic engineering ». In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFCOM)*, p. 1–2.
- Feldmann, A., A. Greenberg, C. Lund, N. Reingold, J. Rexford, et F. True. 2001. « Deriving traffic demands for operational ip networks : methodology and experience », *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, p. 265–280.
- Floyd, S. 2003. RFC 3649 : Highspeed TCP for large congestion windows.
- Floyd, S. et T. Henderson. 1999. « RFC 2582 : The NewReno modification to TCP's fast recovery algorithm ». In *RFC 2582*.
- Floyd, S. et V. Jacobson. 1993. « Random Early Detection gateways for congestion avoidance », *IEEE/ACM Transactions on Networking*, vol. 1(4), p. 397–413.
- Gao, Y., G. He, et J. C. Hou. 2002. « On exploiting traffic predictability in active queue management », In *Proceedings of INFOCOM*, p. 1630 – 1639.
- Gao, Y. et J. C. Hou. 2001. « Analysis and design of an AVQ algorithm for active queue management », *ACM Comp. Comm. Review*, vol. 31(4), p. 123–134.
- Gao, Y. et J. C. Hou. 2003. « A state feedback control approach to stabilizing queues for ECN-enabled TCP connections », In *Proceedings of IEEE INFOCOM*, vol. 3, p. 2301 – 2311.
- Garroppo, R., S. Giordano, M. Pagano, et G. Prociassi. 2008. « On traffic prediction for resource allocation : A Chebyshev bound based allocation scheme », *Computer Communications*, vol. 31, no. 16, p. 3741 – 3751. Performance Evaluation of Communication Networks (SPECTS 2007).
- Ge, A., F. Callegati, et L. Tamil. 2000. « On optical burst switching and self-similar traffic », *IEEE Communications Letters*, vol. 4, no. 3, p. 98–100.
- Gourieroux, C. et A. Monfort. 1990. *Cours de séries temporelles et modèles dynamiques*. Economica.
- Groschwitz, N. K. et G. C. Polyzos. 1994. « A time series model of long-term NSF-NET backbone traffic », In *Proceedings of the IEEE International conference on Communications*.
- He, G., Y. Gao, J. C. Hou, et K. Park. 2002. « A case for exploiting self-similarity of network traffic in TCP ». In *ICNP '02 : Proceedings of the 10th IEEE Interna-*

- tional Conference on Network Protocols*, p. 34–45, Washington, DC, USA. IEEE Computer Society.
- He, G. et J. C. Hou. 2003. « On exploiting long range dependency in measuring cross traffic », *In Proceedings of IEEE INFOCOM 2003, San Francisco, CA*.
- . 2005. « An in-depth, analytical study of sampling techniques for self-similar Internet traffic ». In *ICDCS '05 : Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, p. 404–413, Washington, DC, USA. IEEE Computer Society.
- . 2006. « On sampling self-similar internet traffic », *Journal of Computer Networks*, vol. 50, no. 16, p. 2919–2936.
- He, Q., C. Dovrolis, et M. Ammar. 2005. « On the predictability of large transfer TCP », *SIGCOMM'05, Philadelphia, Pennsylvania, USA*.
- . 2007. « On the predictability of large transfer TCP throughput », *Computer Networks*, vol. 51, no. 14, p. 3959 – 3977.
- Hellerstein, J., Y. Diao, S. Perekh, et D. Tilbury. 2004. *Feedback Control of Computing Systems*. Hoboken, N.J IEEE Press/Wiley-Interscience.
- Hoe, J. C. 1996. « Improving the start-up behavior of a congestion control scheme for TCP ». In *Conference proceedings on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM'96*, p. 270–280, New York, USA. ACM.
- Hollot, C., V. Misra, D. Towsley, et W. B. Gong. 2001. « On designing improved controllers for AQM routers supporting TCP flows », *In Proceedings of IEEE INFOCOM*, p. 1726–1734.
- Hosking, J. R. M. 1981a. « Fractional differencing », *Biometrika Trust - Oxford Journals*, vol. 68, no. 1, p. 165–176.
- . 1981b. « Fractional differencing », *Biometrika*, vol. 68, no. 1, p. 165–176.
- Hunter, D., M. Chia, et I. Andonovic. 1998. « Buffering in optical packet switches », *Journal of Lightwave Technology*, vol. 16, no. 12, p. 2081–2094.
- Internet Protocol Flow Information eXport (IPFIX). <http://ipfix.doit.wisc.edu>.
- Jacobson, V. 1988. « Congestion avoidance and control ». In *SIGCOMM : Symposium proceedings on Communications architectures and protocols*, p. 314–329, New York, USA. ACM.
- Jain, A., A. Karandikar, et R. Verma. 2003. « An adaptive prediction based approach for congestion estimation in active queue management (APACE) », *In Proceedings of IEEE GLOBECOM 2003*, p. 4153 – 4157.

- . 2004. « Adaptive prediction based approach for congestion estimation (APACE) in active queue management », *Computer Communications*, p. 1569 – 1681.
- Janowski, L. et P. Owezarski. 2010. « Assessing the accuracy of using aggregated traffic traces in network engineering », *Telecommunication Systems*, vol. 43, no. 3-4, p. 223–236.
- Jibukumar, M., R. Datta, et P. Biswas. 2008. « Kalman filter based variable bit rate video frame size prediction », *3rd International Symposium on Wireless Pervasive Computing, ISWPC*, p. 459–463.
- Jin, M. et O. Yang. 2006. « On Controlling the Loss Rate in Optical Burst Switching Networks », *3rd International Conference of Broadband Communications, Networks and Systems, (BROADNETS)*, p. 1–9.
- John, W., S. Tafvelin, et T. Olovsson. 2010. « Passive Internet measurement : Overview and guidelines based on experiences », *Computer Communications*, vol. 33, no. 5, p. 533 – 550.
- Jue, J. P. et V. M. Vokkarane. 2005. « Optical burst switched networks ».
- Kalman, R. E. 1960. « A new approach to linear filtering and prediction problems », *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. Series D, p. 35–45.
- Kantz, H. et T. Schreiber. 2006. *Nonlinear Time Series Analysis*. New York, USA : Cambridge University Press.
- Krunz, M. et A. Makowski. 1998. « A source model for VBR video traffic based on M/G/ ∞ input processes ». In *Proceedings of IEEE INFOCOM'98*, p. 1441–1449.
- Larrieu, N. et P. Owezarski. 2005. « Measurement based networking approach applied to congestion control in the multi-domain Internet ». In *IFIP/IEEE International Symposium on Integrated Network Management*, p. 485 – 498.
- Leland, W., M. S. Taqqu, W. Willinger, et D. Wilson. 1994. « On the self-similar nature of Ethernet traffic (extended version) », *IEEE/ACM Trans. Networking*, vol. 2, p. 1–15.
- Leland, W. E. et D. V. Wilson. 1991. « High time-resolution measurement and analysis of LAN traffic : Implications for LAN interconnection », *In Proceedings IEEE INFOCOM*, p. 1360–1366.
- Lim, A. et K. Ab-Hamid. 2000. « Kalman prediction method for congestion avoidance in ATM networks », *Proceedings of TENCON*, vol. 1, p. 346–351 vol.1.
- Lin, W. et R. Morris. 1997. « Dynamics of random early detection », *In Proceedings of ACM SIGCOMM*.

- Lin, Y. et G. Cunningham. 1995. « A new approach to fuzzy-neural system modeling », *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 2, p. 190–198.
- Lin, Y. K. 1976. *Probabilistic Theory of Structural Dynamics*. Robert E. Krieger Publishing Company.
- Lippmann, R. 1987. « An introduction to computing with neural nets », *ASSP Magazine, IEEE*, vol. 4, no. 2, p. 4–22.
- Liu, Z., J. Almhana, V. Choulakian, et R. McGorman. 2006. « A long-range dependent model for Internet traffic with power transformation », *Communications Letters, IEEE*, vol. 10, no. 8, p. 632–634.
- Low, S. H. 2003. « A duality model of tcp and queue management algorithms », *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, p. 525–536.
- Mah, B. 1997. « An empirical model of HTTP network traffic », *Proceedings IEEE INFOCOM*, vol. 2, p. 592–600 vol.2.
- Mandelbrot, B. B. et J. W. V. Ness. 1968. « Fractional Brownian Motions, fractional noises and applications », *SIAM Review*, vol. 10, no. 4, p. 422–437.
- Markopoulou, A., G. Iannaccone, S. Bhattacharyya, C. Chen-Nee, Y. Ganjali, et C. Diot. 2008. « Characterization of failures in an operational IP backbone network », *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, p. 749–762.
- McCanne, S. et S. Floyd. 1995. The network simulator (ns-2). <http://www.isi.edu/nsnam/ns/>.
- Nabe, M. et M. M. H. Miyahara. 1998. « Analysis and modeling of world wide web traffic for capacity dimensioning of Internet access lines », *Perform. Eval.*, vol. 34, no. 4, p. 249–271.
- Nain, P. 1997. *Basic Elements of Queueing theory : Application to the Modelling of Computer Systems*. Lecture Notes.
- Norros, I. 1995. « On the use of fractional brownian motion in the theory of connectionless networks », *IEEE Journal on Selected Areas in Communications*, vol. 13, p. 953–962.
- Norros, I. et P. Pruthi. 1996. On the applicability of Gaussian traffic models.
- Ogata, K. 1997. *Modern Control Engineering*. Prentice Hall, 3rd edition.
- Ott, T. J., T. V. Lakshman, et L. H. Wong. 1999. « SRED : Stabilized RED », In *Proceedings of INFOCOM, New York, USA*, p. 1346–1355.
- Owezarski, P. et N. Larrieu. 2004. « Internet traffic characterization - An analysis of traffic oscillations », 7th *IEEE International Conference on High Speed Networks and Multimedia Communications, Toulouse, France*.

- Packet Sampling (PSAMP) - IETF Working Group. 2009. Packet sampling (PSAMP). <http://datatracker.ietf.org/wg/psamp/charter/>.
- Paganini, F., Z. Wang, S. Low, et J. Doyle. 2003. « A new TCP/AQM for stable operation in fast networks », *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM*, vol. 1, p. 96–105.
- Papagiannaki, K., S. Moon, C. Fraleigh, P. Thiran, et C. Diot. 2003a. « Measurement and analysis of single-hop delay on an IP backbone network », *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 6, p. 908 – 921.
- Papagiannaki, K., N. Taft, Z.-L. Zhang, et C. Diot. 2003b. « Long-term forecasting of Internet backbone traffic : Observations and initial models ». In *IEEE Infocom*, San Francisco.
- . 2005. « Long-term forecasting of Internet backbone traffic », *IEEE Transactions on Neural Networks*, vol. 16, no. 5, p. 1110–1124.
- Parekh, S., N. Gandhi, J. Hellerstein, D. Tilbury, T. Jayram, et J. Bigus. 2002. « Using control theory to achieve service level objectives in performance management », *Real-Time Systems*, vol. 23, p. 127–141.
- Parekh, S., J. Hellerstein, T. S. Jayram, N. Gandhi, D. Tilbury, et J. Bigus. 2001. « Using control theory to achieve service level objectives in performance management », In *Proceedings of the American Control Conference*, p. 3000–3005.
- Park, K., G. Kim, et M. Crovella. 1997. « On the effect of traffic self-similarity on network performance », In *Proceedings SPIE Int. Conf. Performance and Control of Network Systems*.
- Park, K. et W. Willinger. 2000. *Self-Similar Network Traffic and Performance Evaluation*. New York, NY, USA : John Wiley & Sons, Inc.
- Paxon, V. et S. Floyd. 1995. « Wide-area traffic : The failure of Poisson modeling », *IEEE/ACM Trans. Networking*, vol. 3, p. 226–244.
- Paxson, V., A. Adams, et M. Mathis. 2002. « Experiences with NIMI ». In *Symposium on Applications and the Internet (SAINT) Workshops*, p. 108 –118.
- Pletka, R., M. Waldvogel, et S. Mannel. 2003. « PURPLE : Predictive active queue management utilizing congestion information », In *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*.
- Postel, J. 1983. « RFC 879 - TCP maximum segment size and related topics ».
- Qiao, C. et M. Yoo. 1999a. « A novel switching paradigm for buffer-less WDM networks », *Proceedings of Optical Fiber Communication Conference (OFC)*, p. 177–179.

- . 1999b. « Optical burst switching (OBS) - a new paradigm for an optical Internet », *Journal on High Speed Networks (JHSN)*, vol. 8, no. 1, p. 69–84.
- Qiao, Y., J. Skicewicz, et P. Dinda. 2002. « Multiscale predictability of network traffic », *Tech. Rep. NWU-CS-02-13, Department of Computer Science, Northwestern University*.
- Qiao, Y., J. Skicewicz, et P. Dinda. 2003. « An empirical study of the multiscale predictability of network traffic », *In IEEE Proc. of HPDC*.
- Ren, F. Y., C. Lin, et W. D. Liu. 2003. « Congestion control in IP network », *Journal of computers*, vol. 26, no. 9, p. 1025–1034.
- Rosenfeld, F., R. Hannoset, et R. Sabatier. 1988. *Analyse financière et gestion des valeurs mobilières*. T. 1. Dunod Entreprise.
- Rouai, F. A. et M. Ben.Ahmed. 2001. « A new approach for fuzzy neural network weight initialization », *INNS-IEEE International Joint Conference on Neural Network (IJCNN)*, vol. 2.
- Rumelhart, D. E., G. E. Hinton, et R. J. Williams. 1986. « Learning internal representations by error propagation », *Parallel distributed processing : explorations in the microstructure of cognition*, vol. 1 : foundations.
- Sang, A. et S. qi Li. 2000. « A predictability analysis of network traffic », *In Proceedings of IEEE INFOCOM*.
- . 2002. « A predictability analysis of network traffic », *Computer Networks*, vol. 39, no. 4, p. 329 – 345.
- Scherrer, A., N. Larrieu, P. Borgnat, P. Owezarski, et P. Abry. 2006. « Non-Gaussian and long-memory statistical modeling of Internet traffic », *4th International Workshop on Internet Performance, Simulation, Monitoring and Measurements, Austria*.
- Shannon, C. 1948. « A mathematical theory of communication », *Bell System Tech*, vol. 27, p. 379–423, 623–659.
- Shao, Q. et L. Trajkovic. 2004. « Measurement and analysis of traffic in a hybrid satellite-terrestrial network », *Proc. SPECTS 2004, San Jose, CA*, p. 329–336.
- Soule, A., A. Nucci, R. Cruz, E. Leonardi, et N. Taft. 2004. « How to identify and estimate the largest traffic matrix elements in a dynamic environment ». In *SIGMETRICS '04/Performance '04 : Proceedings of the joint international conference on Measurement and modeling of computer systems*, p. 73–84. ACM Press.
- Soule, A., K. Salamatian, A. Nucci, et N. Taft. 2005. « Traffic matrix tracking using Kalman filters », *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 3, p. 24–31.
- Sun, J., K.-T. KO, G. Chen, S. Chan, et M. Zuker. 2003. « PD-RED : to improve the

- performance of RED », *IEEE Communications Letters*, vol. 7(8), p. 406–408.
- Takagi, T. et M. Sugeno. 1985. « Fuzzy identification of systems and its application to modeling and control », *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, p. 116–132.
- Teng, J. et G. N. Rouskas. 2003. « A detailed analysis and performance comparison of wavelength reservation schemes for optical burst switched networks », *Journal of Photonic Network Communications*, vol. 9.
- Turner, J. 1999. « Terabit burst switching », *Journal of High Speed Networks*, vol. 8, no. 1, p. 3–16.
- Vokkarane, V., J. Jue, et S. Sitaraman. 2002. « Burst segmentation : An approach for reducing packet loss in optical burst switched networks », *Proceedings, IEEE, ICC 2002*.
- Vokkarane, V. M., K. Haridoss, et J. P. Jue. 2002. « Threshold-based burst assembly policies for QoS support in optical burst-switched networks », *Proceeding SPIE OptiComm*, p. 125–136.
- Vokkarane, V. M., Q. Zhang, J. P. Jue, et B. Chen. 2002. « Generalized burst assembly and scheduling techniques for QoS support in optical burst-switched networks », *Proceeding of IEEE Globecom*.
- Wang, C., B. Li, K. Shoraby, et Y. Peng. 2003. « AFRED : an adaptive fuzzy-based control algorithm for active queue management », *In Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN'03)*.
- Wang, Y., S. Yu, X. Yang, D. Zou, et Z. Fang. 2006. « H_{∞} -optimal model for VBR video traffic prediction », *IEEE Transactions on Consumer Electronics*, vol. 52, no. 3, p. 1078–1083.
- Welch, G. et G. Bishop. 1995. An introduction to the Kalman filter. Rapport, Chapel Hill, NC, USA.
- Willinger, W., V. Paxson, et M. S. Taqqu. 1998. *Self-similarity and heavy tails : structural modeling of network traffic*. Cambridge, MA, USA : Birkhauser Boston Inc.
- Wolski, R. 1997. « Forecasting network performance to support dynamic scheduling using the network weather service », *In Proceedings of the 6th High-Performance Distributed Computing Conference (HPDC)*.
- Wonnacott, T. H. et R. J. Wonnacott. 1990. « Introductory statistics for business and economics », *Fourth Edition*, Wiley.
- Wu, C. F., K. G. Shi, et D. D. Kandlur. 2002. « The BLUE active queue management algorithms », *IEEE/ACM Transactions on Networking*, vol. 10(4), p. 513 – 528.
- Xu, Y., Z. Wang, et H. Wang. 2005. « ARED : A novel adaptive congestion controller »,

- In Proceedings of the 4th International Conference on Machine Learning and Cybernetics*, p. 18–21.
- Xue, F., J. Liu, Y. Shu, et L. Zhang. 1999. « Traffic modeling based on FARIMA models ». In *In Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, p. 162–167.
- Yu, E. et C. Chen. 1993. « Traffic prediction using neural networks », *IEEE Global Telecommunications Conference, GLOBECOM*, p. 991–995 vol.2.
- Zalesky, A., H. L. Vu, Z. Rosberg, E. W. M. Wong, et M. Zukerman. 2007. « OBS contention resolution performance », *Perform. Evaluation*, vol. 64, no. 4, p. 357–373.
- Zhang, Y. et N. Duffield. 2001. « On the constancy of Internet path properties ». In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement IMW '01* ;, p. 197–211, New York, NY, USA. ACM.
- Zhani, M. F. 2005. *Prévision du trafic Internet Une approche neurofloue*. Mémoire de Mastère en informatique, École Nationale des Sciences de l'informatique.
- Zhani, M. F., H. Elbiaze, et F. Kamoun. 2007. « α -SNFAQM : An active queue management mechanism using neurofuzzy prediction », In *Proc. of the 12th IEEE Symposium on Computers and Communications (ISCC)*.
- . 2008. « Analysis of prediction performance of training-based models using real network traffic », In *Proc. of International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*.
- . 2009a. « α -SNFAQM : An active queue management mechanism using neurofuzzy prediction - extended version », *Submitted to the journal Computer Communications*.
- . 2009b. « Analysis and prediction of real network traffic », *Journal of Networks*, vol. 4, no. 9.
- Zhani, M. F., W. H. Fouad, et H. Elbiaze. 2009a. « On providing QoS in optical burst switched networks using feedback control ». In *34th IEEE Conference on Local Computer Networks (LCN)*, p. 309–312.
- . 2009b. « TCP based estimation method for loss control in OBS networks ». In *Global Telecommunications Conference (IEEE GLOBECOM)*, p. 1 –6.
- Zhijun, F., Z. Yuanhua, et Z. Daowen. 2004. « Kalman optimized model for MPEG-4 VBR sources », *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, p. 688–690.